



Notes from CAST WG Meeting – March 24, 2010  
Test Cell Communications Work Group

Attendees

Joe Barnhart	Roos Instruments
George Brizicky	Teradyne
Andrew Chase	Zenpire Corporation
Valencia dela Vega	Intel
Andrew Gilligan	LTX-Credence
Don Hartman	Freescale
Bill Price	Test Technology Collaboration Consortium
Mark Roos	Roos Instruments
Cathy Rossi-Roos	Roos Instruments
Emilio Salvatoni	Salland Engineering
Joel Taylor	Verigy
Len VanEck	LTX-Credence
Michael Lyman	OptimalTest
Ed Smith	Salvador & Smith Associates
Bill Fritzsche	LTX-Credence
Dean Hedstrom	Aetrium
Kevin Miles	PinTail Technologies
Yota Tsurta	TEL
Sergio Perez	consultant
Rudra Kar	Advantest
Ajay Khoche	consultant
Yuhai Ma	Advantest
Karl Stuber	SEMI

The proposed agenda was as follows:

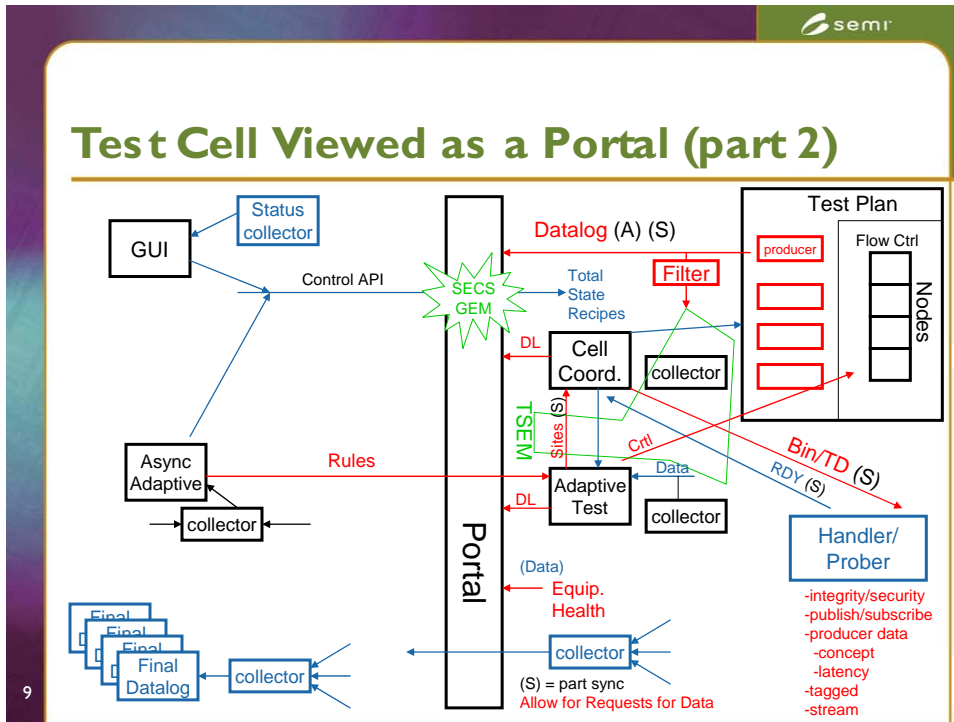
- Clarify the Problem Statement
  - Discussion on use cases – examples
- Review and agree on topology
  - Based on drawing from December meeting
- Discuss alternate specifications
  - Ramifications on recommended specs

The problem statement discussion yielded the following:

At a high level, problem statement is, how do we get equipment in a test cell to work together seamlessly? The end, top-down goal is to make a test cell “plug and play”, meaning, equipment can be placed in a cell, a cell coordinator configures the test cell and it’s ready to test. The other part of the discussion was a bottom-up approach which discussed more practical idea on how to reach this goal. The remainder of the meeting was focused on the bottom up approach.

Topology

It was generally agreed that the diagram derived from the December 16 2009 meeting was a good starting point for a topology of the test cell. The diagram is presented below. From a topology point of view, the left hand side of the diagram represents asynchronous data transaction – those transactions not needed for immediate run-time decisions. The right hand site is considered “part synchronous” and data transactions are more real-time as decisions need to be made immediately (pass/fail, binning etc.).



One item that came out of the topology discussion was the need for a dictionary of definitions – this was viewed as necessary to insure everyone was “on the same page” when a specific term was used in discussion. The dictionary developed via this discussion appears below:

#### Test Cell Communication Work Group – Dictionary

Data warehouse – all of the data, all of the time. Represents the accumulation of all data generated by the test cell and all the data is accessible from this area. The function is asynchronous in nature.

- Future consideration – splitting the function into async data, near-time data etc.

Data Collector – the purpose is to collect streams of data and consolidate into a composite stream. Multi-input, single output stream.

Data Producer – one end of a unidirectional stream

Data Consumer – terminating end of a unidirectional stream.

Filter – could follow a collector or precede a collector. Can only act on one stream at a time - one input stream, one control stream, one output stream.

Graphical User Interface (GUI) – could be one of several interfaces (Operator, engineering, maintenance etc.). In order to perform several function, it needs to be connectable and supported by a protocol. GUI would manage:

- Monitoring
- Control
- Configuration

Control API – ability to control and find out what is happening on any piece of equipment in the test cell.

Asynchronous Adaptive – defined as the floor side of the adaptive test function. Potentially could take data from several cells in order to decide on new rules to pass on to the cell side of the adaptive test function

Adaptive Test – defined as the cell side of the adaptive test function. Provides real-time decision on flow with input from the floor side and from data immediately available on the cell side.

NOTE: Adaptive test is used as a specific example of a more generic concept where there is distributed function on the floor and in the cell.

Portal – further discussion required

- Initially a demarcation line, a logical separation between floor and cell.
- Could contain additional functions – security, isolation, firewall functions.

Cell Coordinator – directory for the cell – location where other functions look up information and holds the cell configuration data. Also provides group synchronization – insures that all functions occur in proper sequence.

Use Model Discussion

The general topology agreement led to the notion of services and which services each party in the cell needed or could supply (tester, handler/prober, cell coordinator and adaptive test functions for now).

This part of the discussion started with a use model – a final test lot of device arriving at a cell – what happens from a setup, start testing and end of lot point of view. The initial premise was to describe the current cell functionality in light of the new architecture (i.e. what services do we need to define to describe the current cell functionality) This was viewed as a necessary step to gradually move into new specification and maintain compatibility with the current legacy setups. This will set the stage for advancement in the direction of the new architecture. It was stated that this new direction will, in all likelihood, be driven by adaptive test.

- I. Request to test in a given cell occurs. Can be a bar code read or manual request.
  - a. How do we insure that all equipment is in the cell at start time?
    - i. Everything needs to have a unique name to configure the cell
    - ii. Association with the cell coordinator discovers what each piece of equipment can do.
  - b. Who does the discovery?
    - i. Need to resolve the physical location issue.
  - c. Each piece of equipment needs to define what information/data it needs
    - i. Handler
      1. respond to requests
      2. binning service
    - ii. Tester
    - iii. Cell Coordinator
    - iv. Adaptive Test Function
    - v. Station Controller
  - d. Grade the equipment in the cell entities
    - i. What level of the spec is supported

For Next Meeting

- I. Attempt to define what core set of services are needed between pieces of equipment.
  - a. Version 1 – “code” the current functionality of a test cell in terms of the new architecture.
  - b. Next version – speculate on what new message/requests need to be added in terms of service additions or modifications.