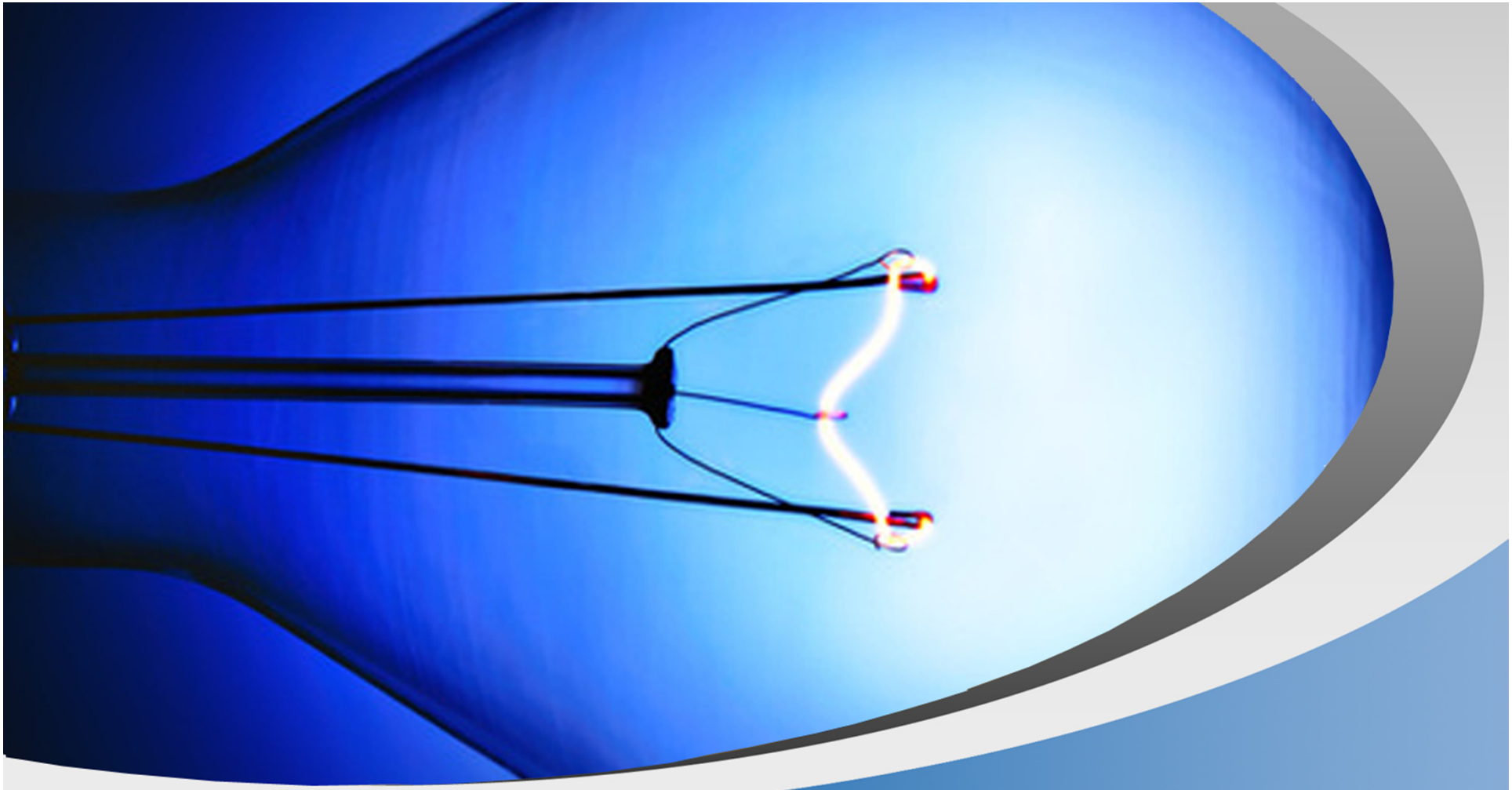
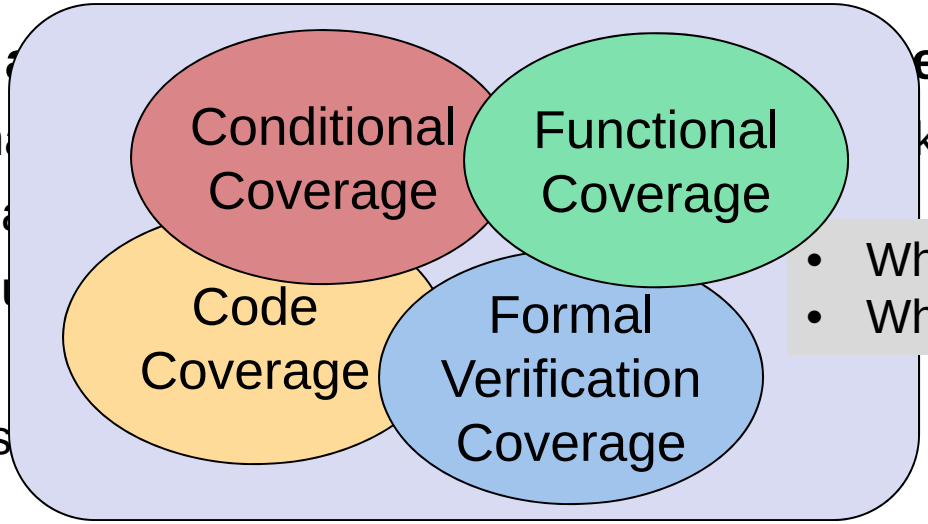


An Introduction to the Unified Coverage Interoperability Standard – UCIS Technical Committee

accellera
SYSTEMS INITIATIVE

Motivation for UCIS

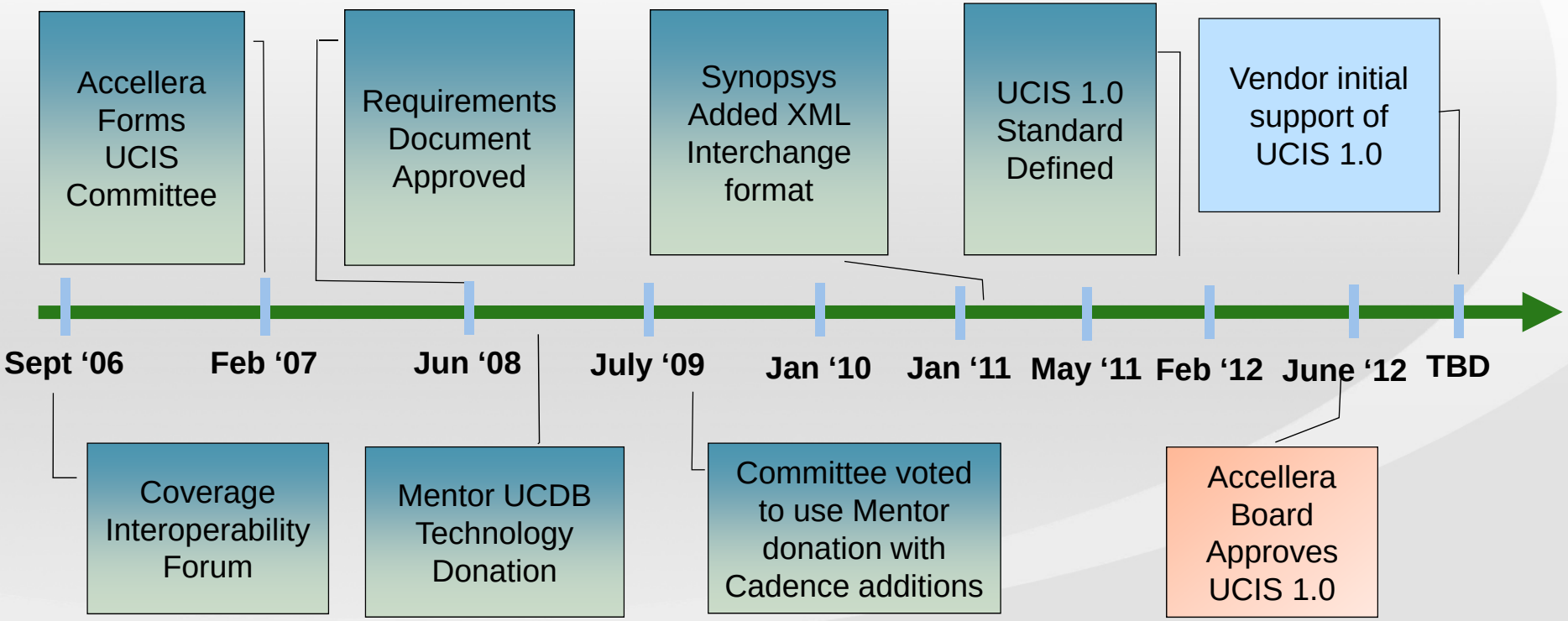
- **Verification is hard**
  - <insert standard slide: 70+%, increasing complexity, yadda, yadda, yadda>
- **Variety of verification techniques and methods**
  - Directed and constrained-random simulation
  - Formal verification
  - Testbench methodologies
- **Design a** **erage metrics:**
  - What ha ked?
  - How ma
  - How mu
  - Where
  - What is problem areas?



- What coverage overlaps?
- What coverage is missing?

accellera SYSTEMS INITIATIVE

UCIS Development Timeline



- Accellera UCIS Committee Deliverables to Member Companies
  - UCIS Specification V1.0
  - UCIS API Header File (.h) for member companies
  - XML template

accellera SYSTEMS INITIATIVE

UCIS Contributors

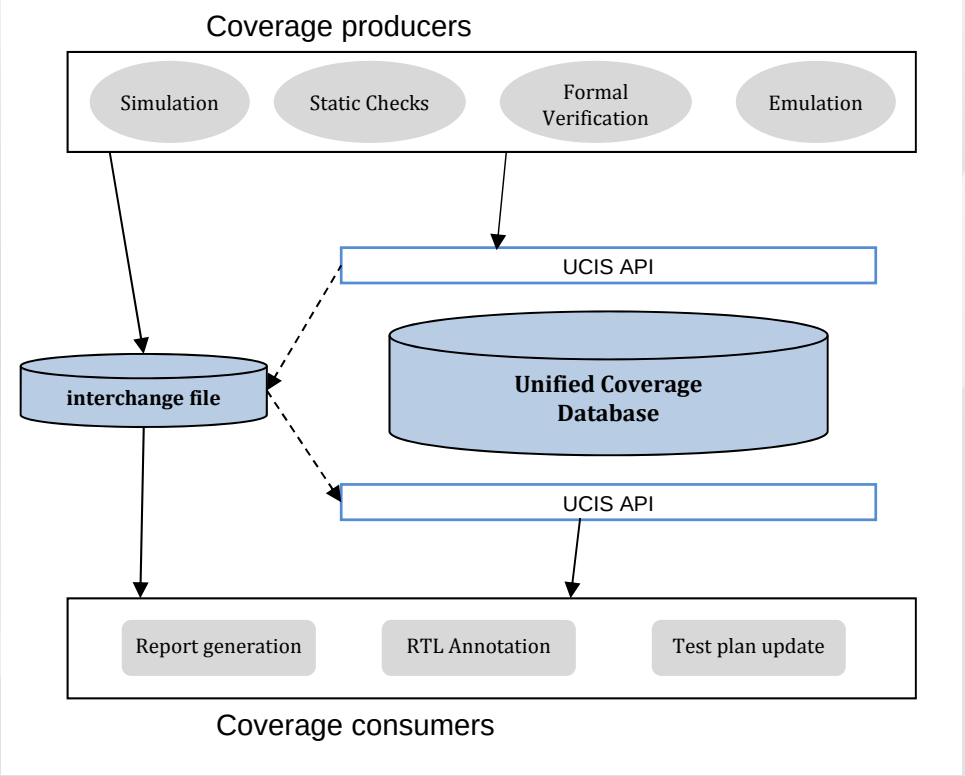
- **Cadence Design Systems**
  - Hemant Gupta
  - Sandeep Pagey
  - John Brennan
- **Carbon Design**
  - David Scott
- **Cisco Systems**
  - Mark Strickland
- **Freescale Semiconductor**
  - Amol Bhinge
  - Hillel Miller
  - George Wood
- **IBM**
  - Nancy Pratt
- **Jasper Design Automation**
  - Rajeev Ranjan
- **Mentor Graphics**
  - Abigail Moorhouse
  - Samiran Laha
  - Andrew Seawright
- **Oracle Corporation**
  - Michael Burns
- **Synopsys Inc.**
  - Vernon Lee
  - Surrendra Dudani
  - Dan Benua
  - Mehdi Mohtashemi
- **Paradigm-Works**
  - Ambar Sarkar
- **D. E. Shaw Research**
  - Richard Ho

Agenda

- Introduction
- Use Cases and Data Flow 
- UCIS Data Model
- XML Interchange Format
- Formal
- Limitations
- Wrap-up
- Q&A

Unified View of Coverage Data

- **Provides a standard API for tools that produce and consume verification coverage data**
  - Multiple heterogeneous coverage data producers
  - Multiple heterogeneous coverage data consumers
    - Reporting tools
    - Analysis tools
    - Test plan tools
    - Advanced verification tools



accellera SYSTEMS INITIATIVE

Coverage Flow & Associated Tools

- **Generate coverage** →
  - Single verification run, single/multiple coverage types
  - Multiple verification runs
- **Access coverage** →
  - Using UCIS Application Programming Interface (API)
  - Using Interchange Format (XML Interchange Format)
- **Merge coverage** →
  - Across runs, components, tools
- **Analyze coverage** →
  - Report unhit coverage points
  - Track progress of coverage over time

Simulation	Register
Formal	HW/SW
Acceleration	FPGA Prototype
Emulation	Systems Tools

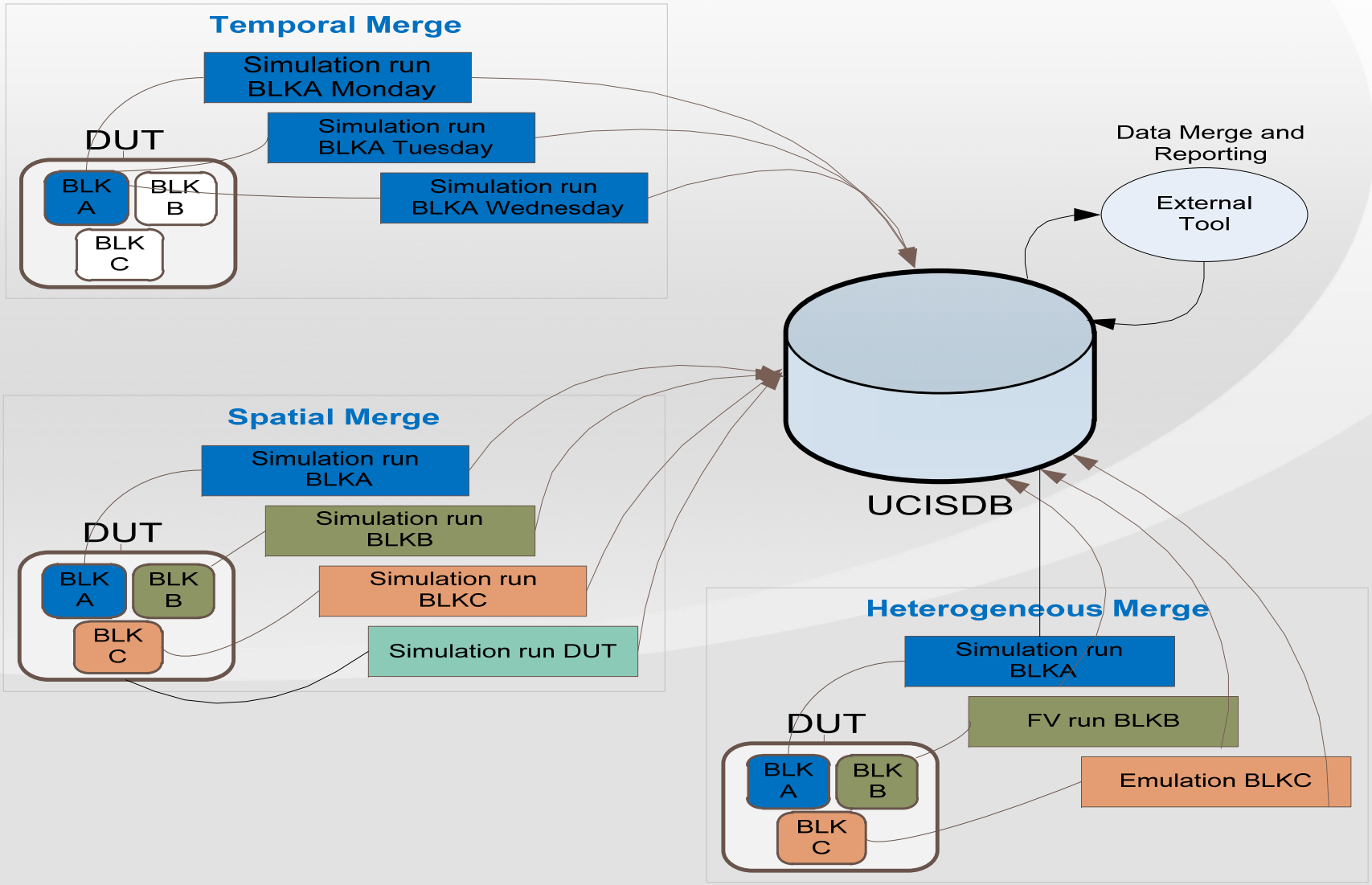
UCIS API	UCIS XML
- Read	- Open
- Write	- Write

Scope of UCIS

Coverage Analysis Tools
Customers Management Tools
Vendors Management Tools

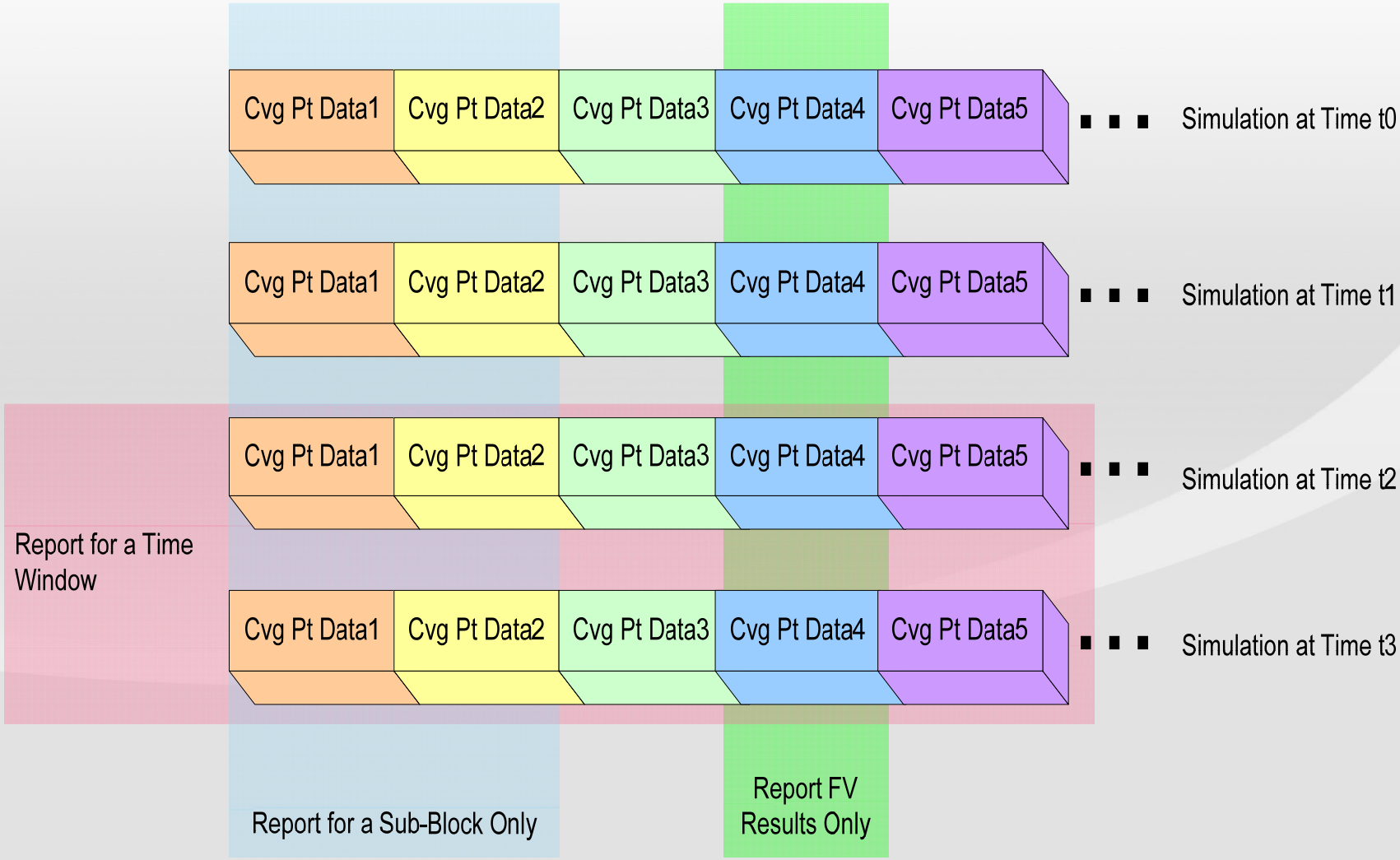
accellera SYSTEMS INITIATIVE

Use Cases and Data Flow - Merge



accellera SYSTEMS INITIATIVE

Use Cases and Data Flow - Report



accellera SYSTEMS INITIATIVE

Agenda

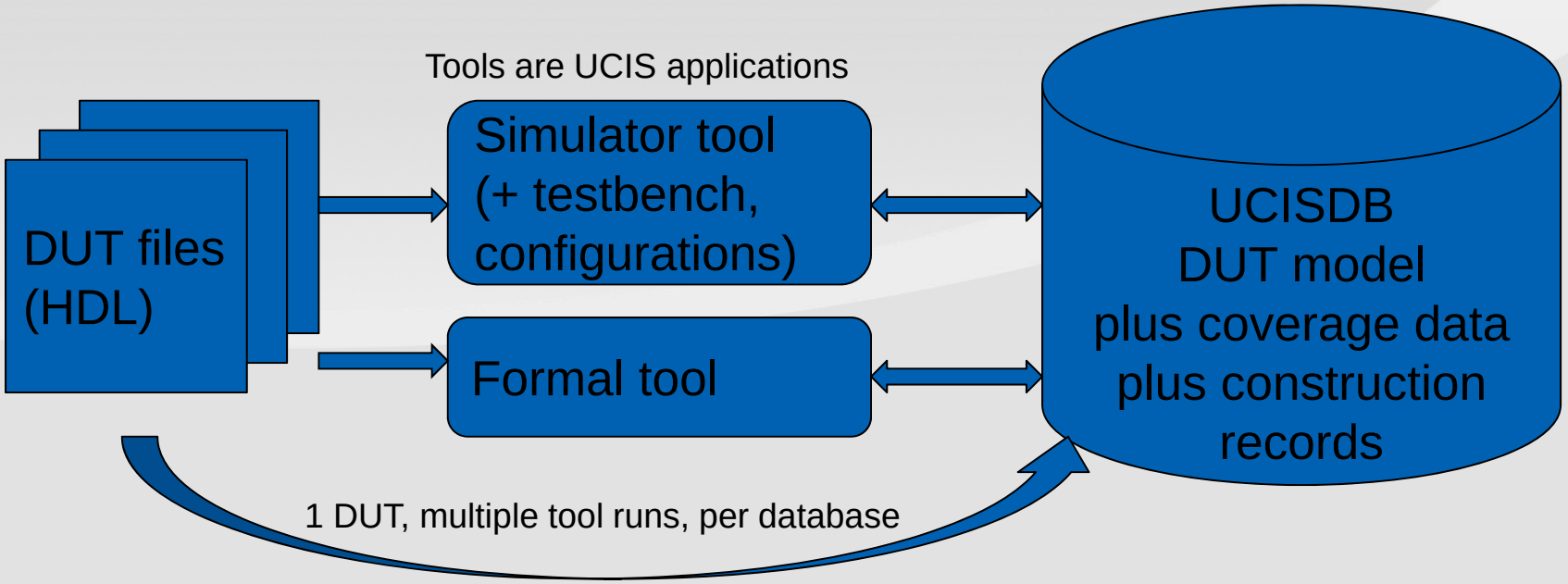
- Introduction
- Use Cases and Data Flow
- UCIS Data Model 
- XML Interchange Format
- Formal
- Limitations
- Wrap-up
- Q&A

accellera SYSTEMS INITIATIVE

Universally Recognizable and Accessible Coverage Data

(the UCIS mission statement)

- **Coverage data – a statement about the target domain**
- **Accessible – a statement about the new API and implementations**
- **Universally recognizable – a statement about the data**

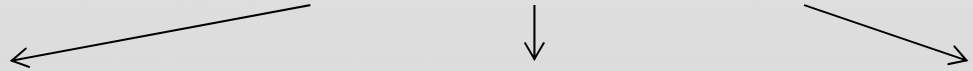


accellera SYSTEMS INITIATIVE

Coverage Data

- The Information Model tells us how to represent coverage data
- Highly generalized model of coverage may be stated as

@event if (condition) counter++



- Variable value change
- Statement execution
- Covergroup sample

- Sample value
- Disabled

- SV bins
- Assertion status
- Any count!

Same event

Different conditions

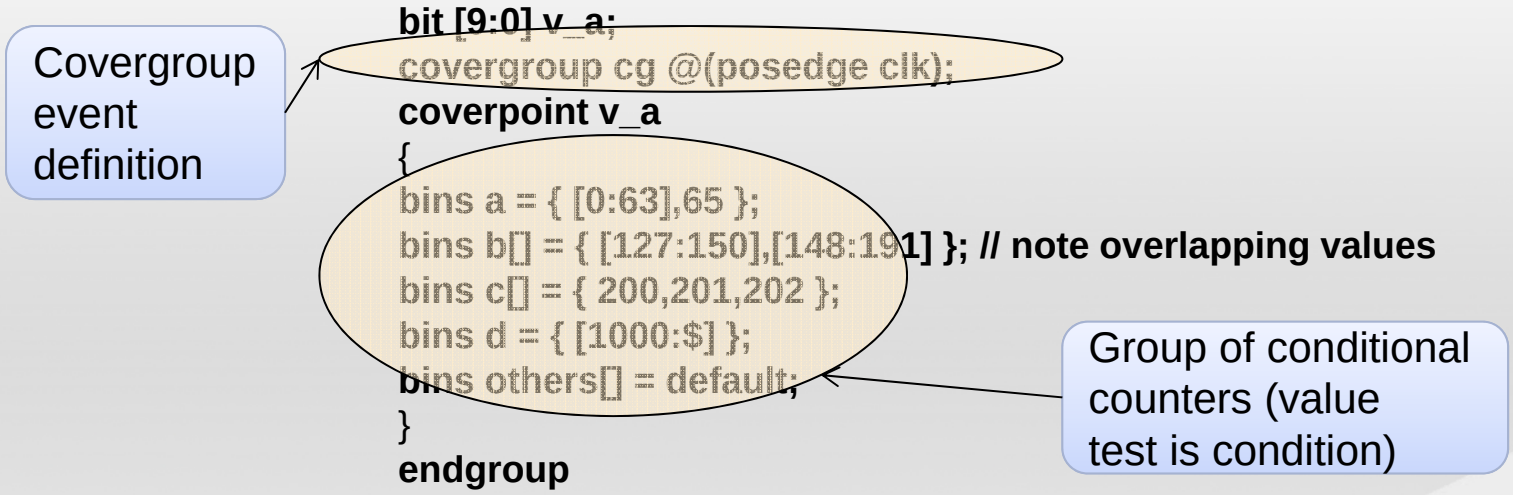
Multiple 'associated' counters

accellera SYSTEMS INITIATIVE

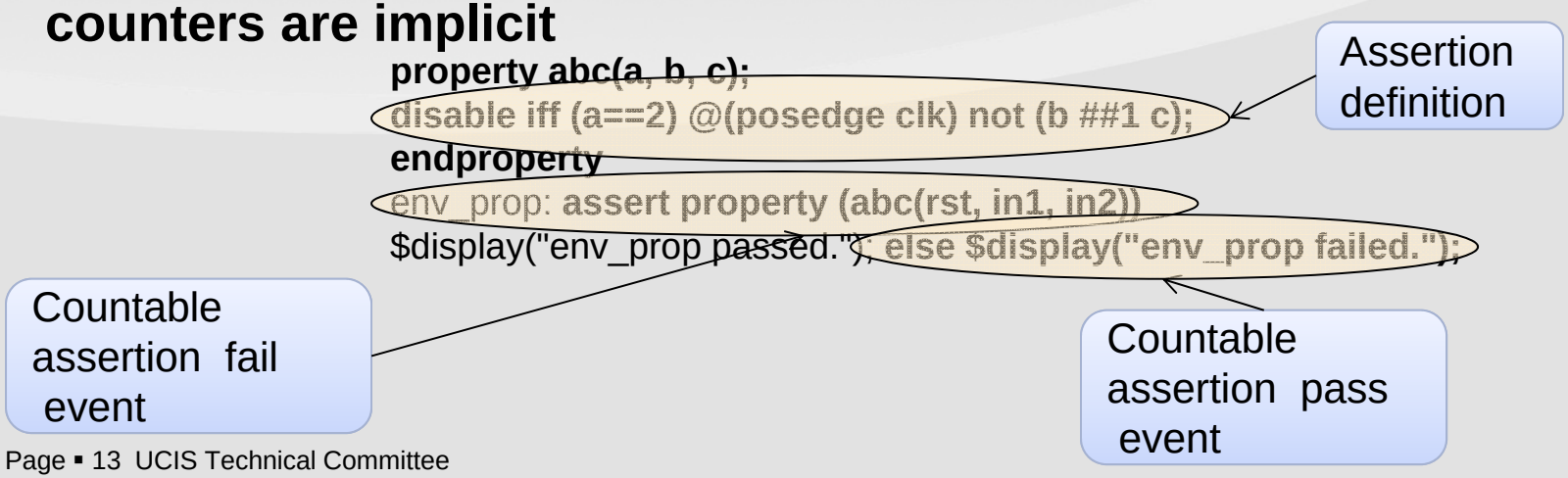
Information Model Analysis

(the search for identifiable events that are countable and groupable in condition sets)

- **SV Covergroups –define constructs that closely match general model**



- **Assertions – 'negative' and optional 'positive' count semantics, counters are implicit**



accellera SYSTEMS INITIATIVE

Data Model

- **A common data model is defined to hold the range of coverage information models**
  - An API provides write/read access to this data model
  - The **data model ~~=~~ information model**, but can represent a useful subset of it.
- **The data model uses a PRIMARY KEY lookup method to underpin unambiguous object recognition**
- **The UCIS data model has these primary representational forms**
  - Scopes – composite objects to represent structure for HDL design units, instantiated HDL objects and coverage aggregations
  - Coveritems – leaf composite objects to represent the information model (counters)
  - History Nodes – composite objects to hold meta data about the environment, configuration, and other tool and collection processes
  - Attributes, tags, and flags – decoration on the composite objects to add meaning to them

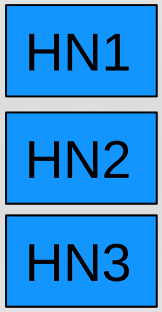
UCIS Object Types

- **Scopes**
  - Must have a name and a scope type.
  - May be a component of a scope hierarchy (scopes may own zero or more child scopes and zero or more coveritems)
  - The combination of name and type is the primary key for the scope and must be unique under the parent. This is a data model requirement.
- **Coveritems**
  - Must have a name and a coveritem type.
  - A leaf node and may not own scopes or coveritems
  - The combination of name and type is the primary key for the coveritem and must be unique under the parental scope.
- **History Nodes**
  - Have a logical name which is their primary key, unique within a UCISDB
  - Record the test contribution history to the database

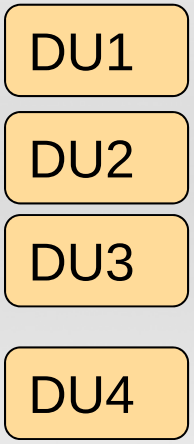
UCIS API - Interface to the Data Model

- **Generic view of a UCIS Database**

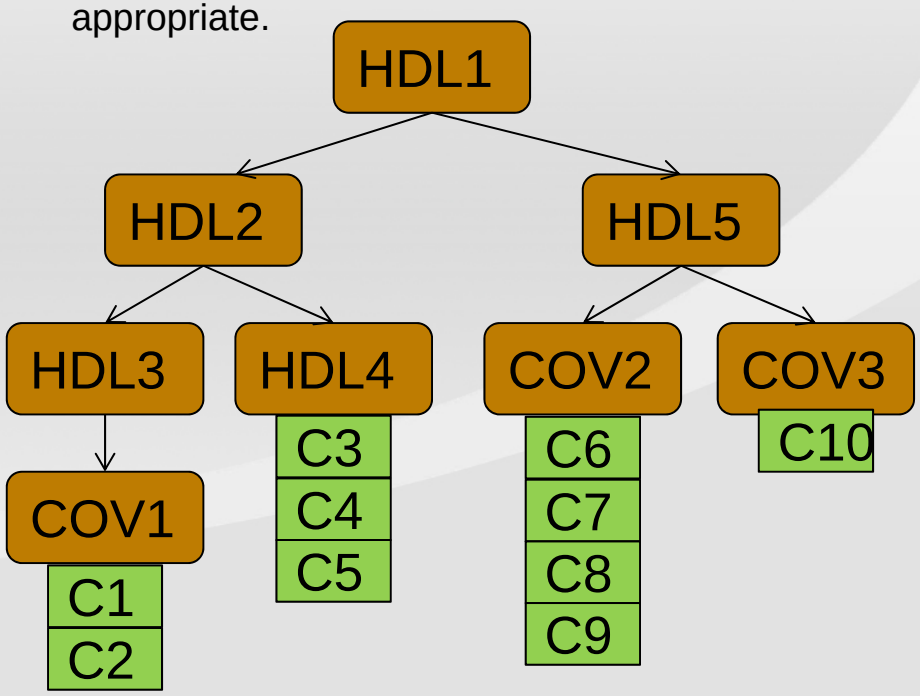
List of history nodes to record database construction



List of scope constructs describing the design units



Hierarchy of scope and coveritem constructs representing the instantiated DUT, plus additional coverage constructs where appropriate.

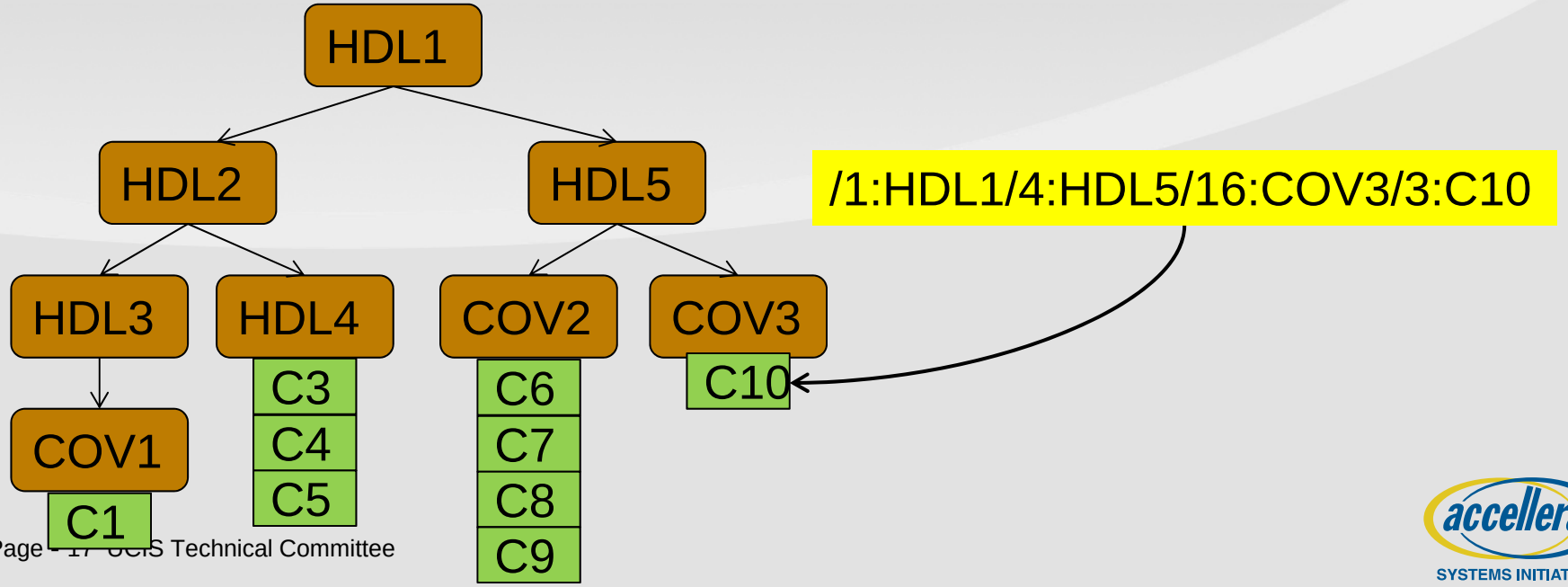


- **The API routines are used to find, create, query, explore, annotate…**

accellera SYSTEMS INITIATIVE

Unique IDs

- **Unique IDs, as the name implies, uniquely identify a single scope or coveritem in the UCISDB**
- **They rely on the defined primary key and structural behavior, specifically that the type/name combination of each node is unique under its parental scope.**
- **The Unique ID is therefore constructed from a list of primary keys starting at the top of the hierarchy and tracing the path to the target**

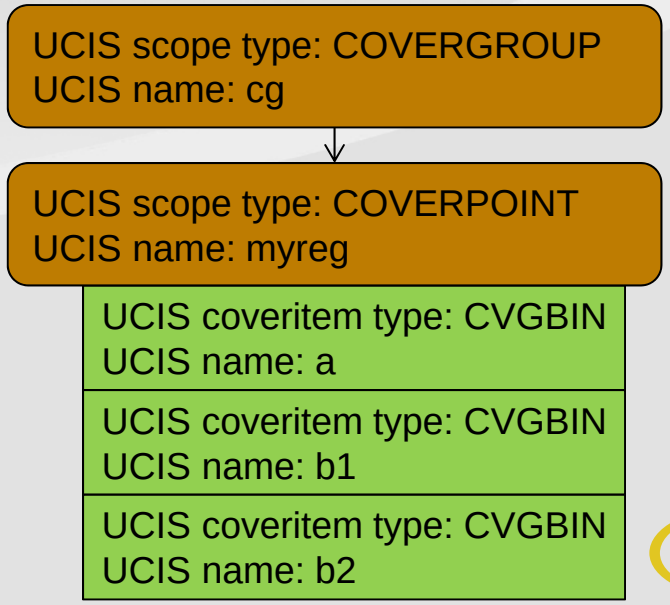


accellera SYSTEMS INITIATIVE

Universal Object Recognition

- **Data is mapped from the information model to the data model**
  - Needs agreement on how unique data in the information model is mapped to data model -- expressed as advisory templates for interpreting data
  - Both structural and naming features are necessary to map complex models
  - Other models are not precluded by the standardized templates
  - Standardized mappings make objects universally recognizable
- **Universal recognition is not canonical naming of coverage items**
  - Primary key management makes objects unambiguous

```
bit [9:0] myreg;
covergroup cg @(posedge clk);
  myreg: coverpoint myreg
  {
    bins a = { [0:63],65 };
    bins b[] = { [127:150],[148:191] };
  }
endgroup
```



accellera SYSTEMS INITIATIVE

Agenda

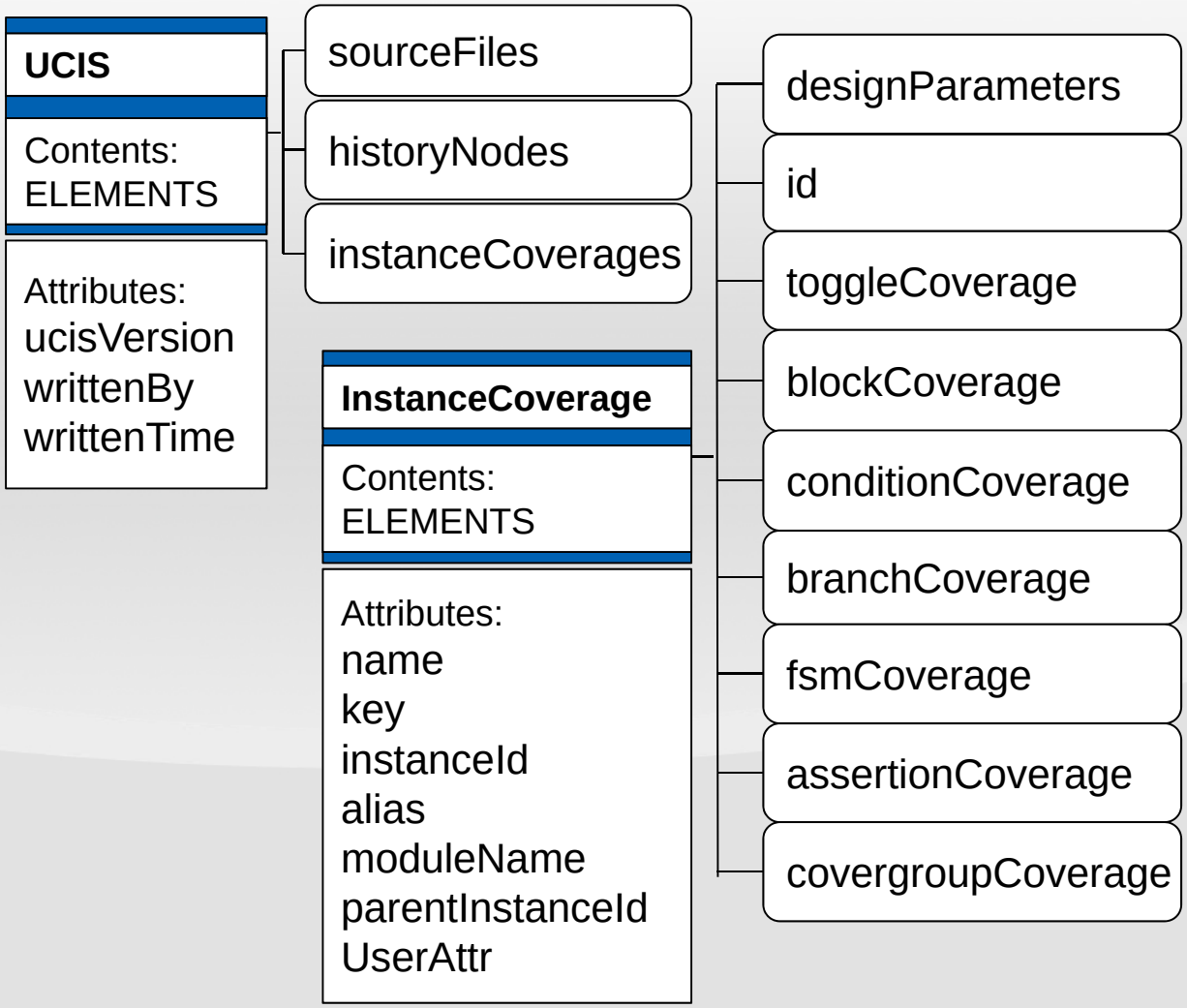
- Introduction
- Use Cases and Data Flow
- UCIS Data Model
- XML Interchange Format 
- Formal
- Limitations
- Wrap-up
- Q&A

accellera SYSTEMS INITIATIVE

Interchange Format (XML)

- **Interoperability requires exchange of coverage data between vendors and users**
  - Exchange data in a portable way
  - XML is a hardware and software independent tool for transporting data
- **Interchange format enables exchange of data between environments**
  - Dump coverage data from one environment and read in another
  - Dump data from multiple environments and merge into another
  - Read data from one environment, modify, then read into another
- **XML schema for the interchange format**
  - Follows the UCIS coverage model
  - Defines the tags used in the interchange format
  - Includes some of the semantic rules of the coverage model

Top level XML Schema element



XML schema defines

- Elements: tags that **must** appear
- Attributes: tags that are **optional**
- Element hierarchy and its child elements
- Number & order of child elements
- Data types for elements & attributes
- Define default & fixed values for elements & attributes

accellera SYSTEMS INITIATIVE

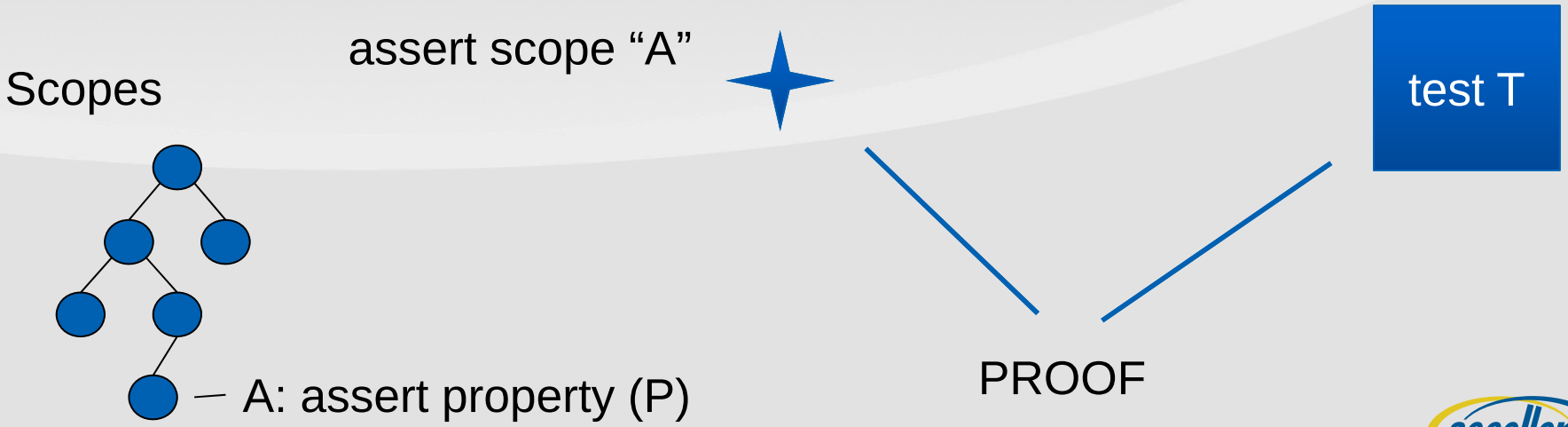
Agenda

- Introduction
- Use Cases and Data Flow
- UCIS Data Model
- XML Interchange Format
- Formal
- Limitations
- Wrap-up
- Q&A

accellera SYSTEMS INITIATIVE

UCIS Assertion Formal Status API

- `ucis_SetFormalStatus(db, test, assertscope, status )`
  `ucis_GetFormalStatus(db, test, assertscope, &status)`
- **status can be:**
  FAILURE, PROOF, VACUOUS, INCONCLUSIVE, ASSUMPTION
- **Similar APIs for:**
  - Setting location of witness waveforms
  - Formal radius (for INCONCLUSIVE, FAILURE)



accellera SYSTEMS INITIATIVE

Formally Unreachable Coverage Items

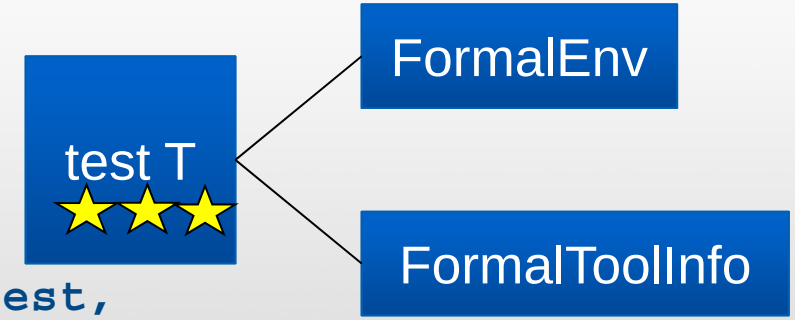
- **Formal tools can mark formally reachable coverage items using the usual simulation APIs**
- **An additional formal specific UCIS API is defined for identifying *formally unreachable* coverage items**
- **Any coverage item can be marked unreachable with respect to formal test(s) (not just assertion items)**


```
ucis_SetFormallyUnreachableCover Test(db, test,
                                      coverscope, index)
ucis_GetFormallyUnreachableCoverTest(db, test,
                                      coverscope, index, &flag)
```


accellera SYSTEMS INITIATIVE

Formal Tests and Formal Environment

- **What makes a test a formal test?**
  - A formal test is a UCIS test with associated formal data




```
ucis_FormalTestGetInfo(db,test,
                &toolInfo,&formalEnv,&coverageContext);
```

- **Basic test info (same as simulation tests, username, run date, etc.) properties on test ucisHistoryNode object** ★
- **Formal Environment object – generic formal info**
  - assumptions used, and scope of analysis – what part of the design was formal run on?
- **Formal tool specific info - FormalToolInfo**
  - setup file, report file, result database.
- **Coverage context – a string property which describes how the coverage is to be interpreted**

accellera SYSTEMS INITIATIVE

Agenda

- Introduction
- Use Cases and Data Flow
- UCIS Data Model
- XML Interchange Format
- Formal
- Limitations
- Wrap-up
- Q&A

accellera SYSTEMS INITIATIVE

Limitations (caveat emptor)

- **Information loss is inevitable**
  - Data transfer to the UCISDB data model is necessarily selective
  - Data is retained to support common coverage models
- **There is an assumption that the original source code is available to the engineer interpreting the UCISDB.**
  - Source code is not transferred to the UCISDB
- **Interchange of data between tools**
  - Standard API does not automatically equal interoperability
  - Many metrics modeled differently by vendors
  - User-supplied names may be necessary
    - E.g. covergroups, class instances
- **High-level operations implemented as user applets**
  - Merging
  - Filtering
  - Calculating Coverage Scores
  - Test ranking
  - Heterogeneous merge

EDA Industry Support

The Unified Coverage Interoperability Standard has been contributed by and developed by primary suppliers in EDA. All of the vendors listed here have donated substantial time to achieve a 1.0 draft, and plan on initial tool support starting in 2012.

accellera SYSTEMS INITIATIVE

UCIS Roadmap

- **With UCIS 1.0, the focus is on user adoption and feedback.**
  - Forward looking roadmap will be prioritized and driven by the user community
- **Adoption and Feedback**
  - Open, user-derived collection of applications
  - Channel to collect feedback and feature requests
    - mailto: review-ucis@lists.accellera.org
  - Discussion forum to tackle current open issues
- **Information about UCIS is available at:**

http://www.accellera.org/activities/committees/ucis

Q&A

accellera SYSTEMS INITIATIVE