An AP210-based PCA/PCB DFx analysis tool

A. Seth, D. Mukhopadhyay, D. Tang, P.M. Ferreira,
University of Illinois at Urbana-Champaign

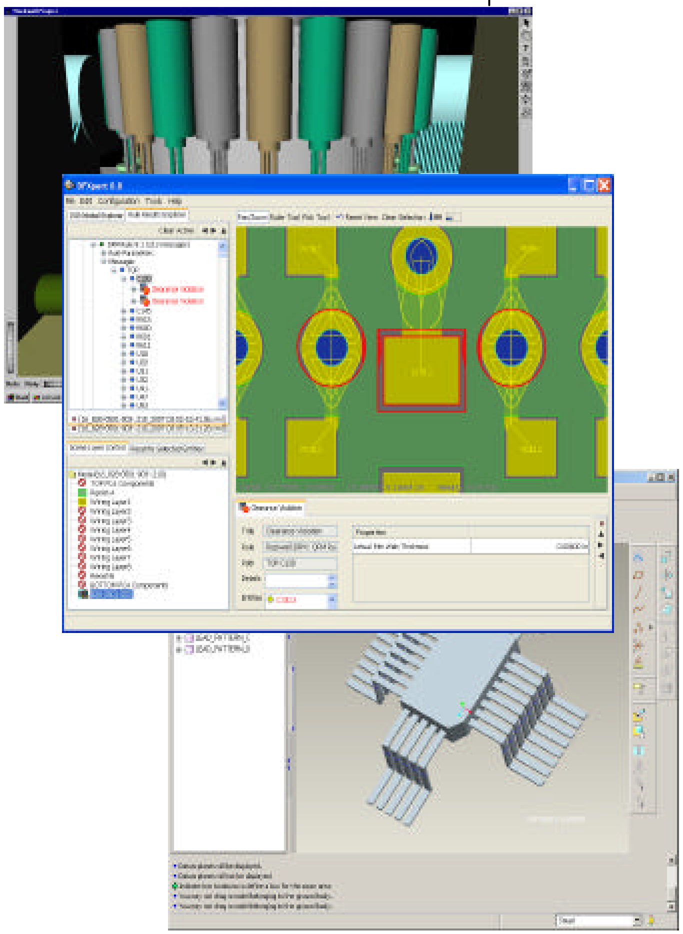
T. Thurman, Rockwell Collins, Inc.

J. Stori, SFM Technology, Inc.

NASA-ESA PDE 2007
May 4, 2007

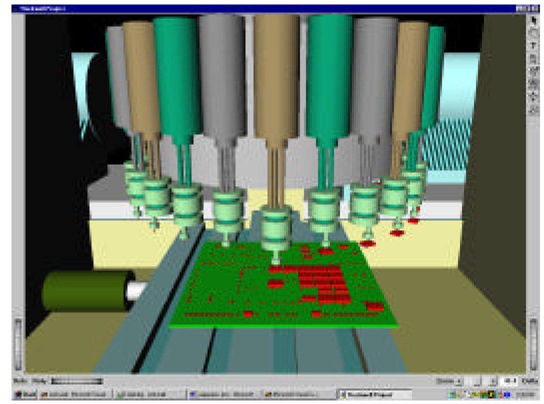
Background

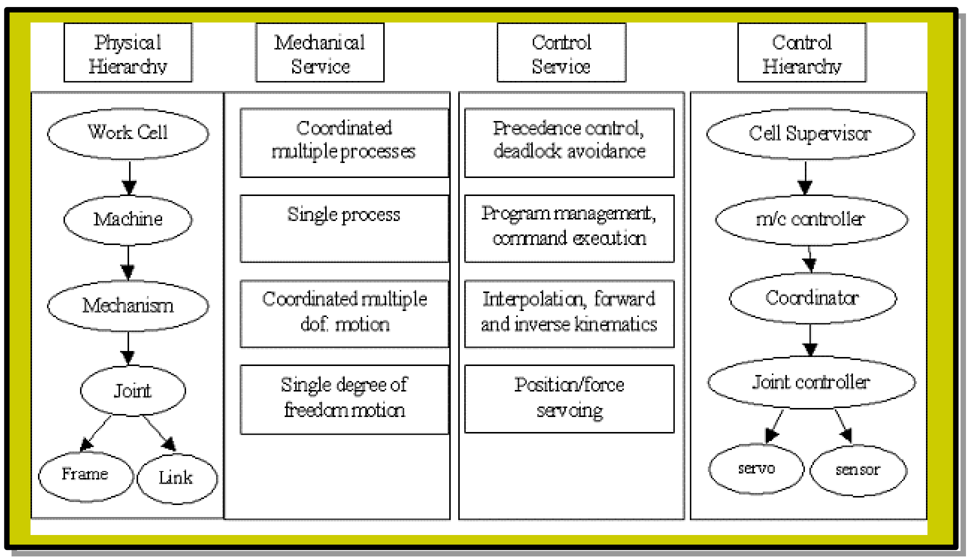
- DFx tool resulted from a research relationship between RCI and UIUC
  - AP210 viewer, geometry library, and rule-based analysis
  - System simulation and control
  - Package modeler (funded by NIST)
- SFM Technology, Inc. acquired a license from UIUC to commercialize the DFX technology in Fall '05
  - Supporting the pilot and production deployment of the tool at RCI
  - Generalizing and enhancing the capabilities for use by RCI and others



Related UIUC Research – System Modeling, Planning, and Control

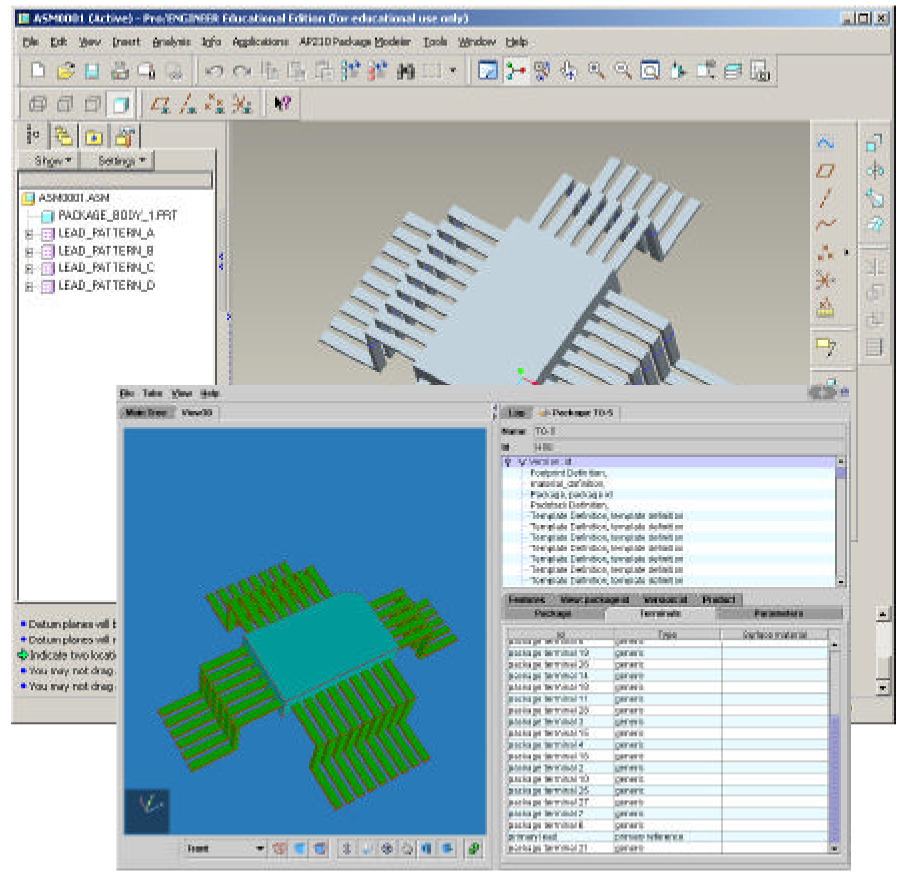
- Hierarchical synthesis of systems for planning, simulation, and control
- Hybrid supervisory control of discrete event systems





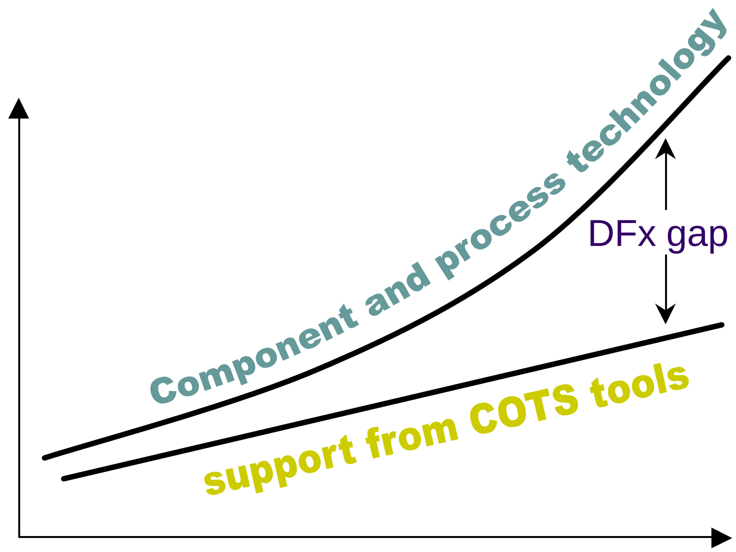
Related UIUC Research – AP210 Package Modeler

- Support the creation of an AP210 packaged component library using a commercial CAD tool.
- Parametric model definition for rapid reconfigurability
- Population of body and lead geometry, seating plane, component footprint, etc.



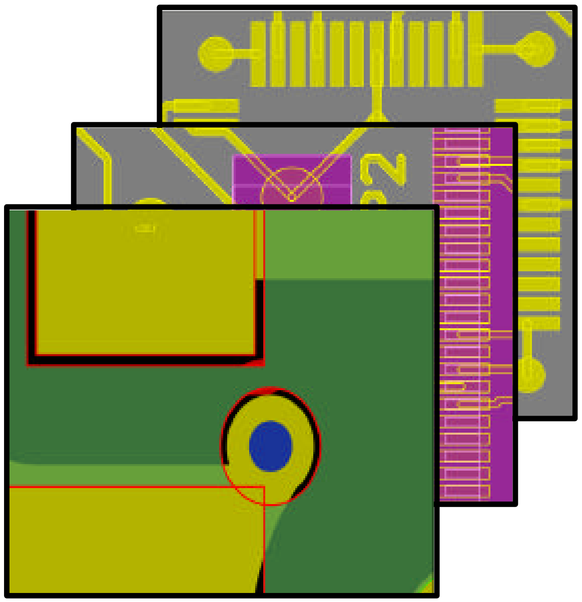
Why DFX?

- Industry trends towards increasing PCA/PCB complexity, product customization, decreased development cycle times, and cost pressures
  - Low-volume production further increases challenge
- It has been estimated that:
  - 60% of overall product cost is determined by decisions made early in the design process
  - 75% of manufacturing cost is determined by design drawings and specifications
  - 70-80% of all product defects are directly related to design issues



Motivation

- Develop a flexible tool for high-value DFX analysis
  - Leveraging Rockwell Collins / UIUC research efforts and strengths of the AP210 representation
  - Build on expertise in manufacturing, geometric computation, process planning
  - Adopt open standards and software technologies (Java, JSDAI, XML, webservices)
  - Provide an accessible interface for a wide-range of application end-users (designers, test and producibility engineers, quality and management personnel)
- Scope
  - DFA (Component analysis, Fiducials, Padstack analysis, Solder paste)
  - DFM (Minimum etch spacings, Solder mask analysis, Holes, vias, microvias)
  - DFT (ICT and FP Test pad selection, Orientation, Inspection)



DFX analysis based on open standards

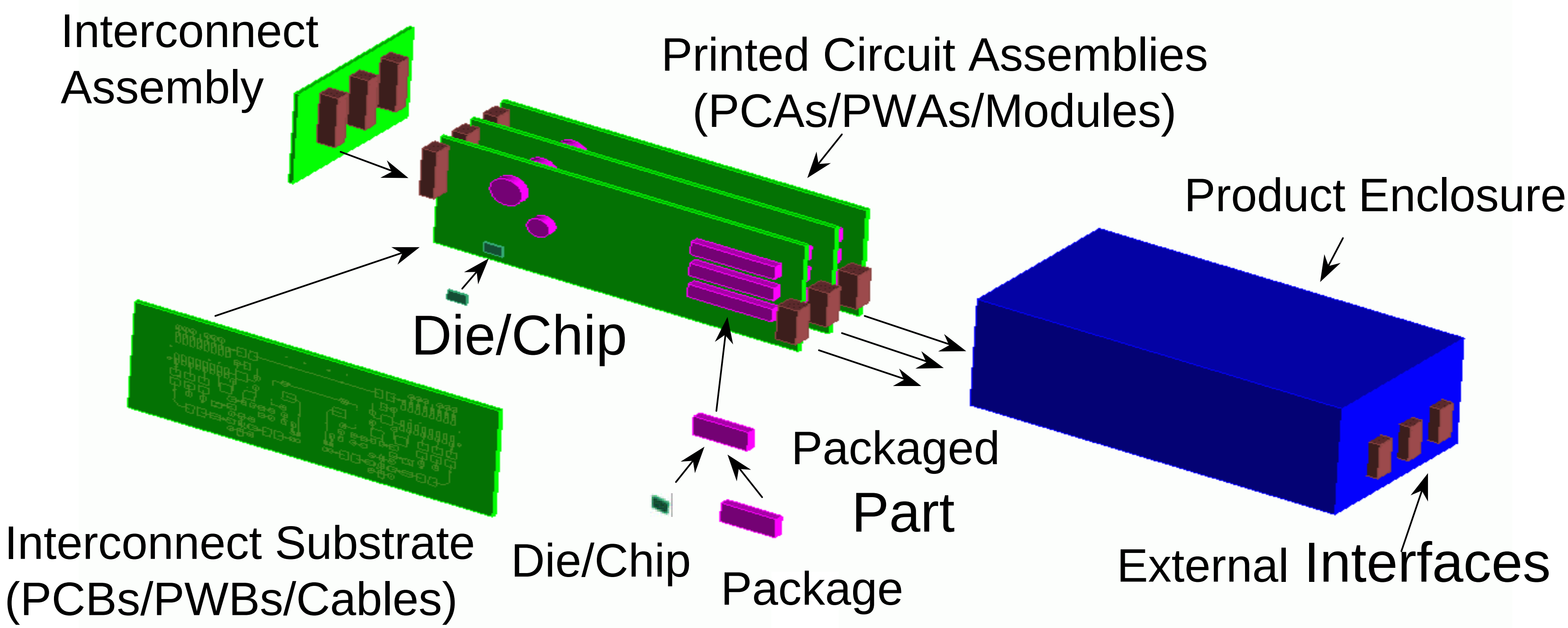
- Compliant with the STEP-AP210 international (ISO) standard for product data exchange and representation
- What is STEP?
  - ISO 10303 is an International Standard for the computer-interpretable representation of product information and for the exchange of product data.
  - The objective is to provide a neutral mechanism capable of describing products throughout their life cycle. This mechanism is suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases, and as a basis for archiving.

STEP Application Protocols (APs) include:

- AP202 - Associative draughting
- AP203 - Configuration controlled 3D design
- AP209 - Composite and metallic structural analysis and related design
- AP210 - Electronic assembly, interconnect and packaging design
- AP212 - Electrotechnical design/installation
- AP214 - Automotive mechanical design processes
- AP232 - Technical data packaging
- AP233 - Systems engineering
- AP239 - Product life cycle support (PLCS)
- AP219, 224, 238, 240 ... Manufacturing ...
- AP215, 216, 218 ... Ship ...
- Others: Building, Piping, Furniture, Oil&Gas, ...

AP 210 Domain

Configuration Controlled Design and Use of Electronic Assemblies, their Interconnection and Packaging



AP 210 Scope

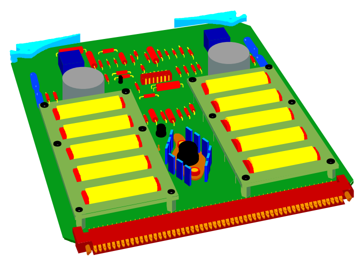
Functional Models
- Functional Unit
- Interface Declaration
- Network Listing
- Simulation Models
- Signals
- Test Bench

Requirements Models
- Design
- Constraints
- Interface
- Allocation

Component / Part Models
- Analysis Support
- Package
- Material Product
- Properties
- “White Box”/ “Black Box”
- Test Bench

Rules Models
- Design
- Manufacturing
- …

Assembly Models
- User View
- Design View
- Component Placement
- Material product
- Complex Assemblies with Multiple Interconnects



Interconnect Models
- User View
- Design View
- Bare Board Design
- Layout templates
- Layers

Design Control
- Geometric Dimensioning and Tolerancing

Configuration Mgmt
- Identification
- Authority
- Effectivity
- Control
- Net Change

Geometric Models
- 2D
- 3D
- CSG, Brep…
- EDIF, IPC, GDSII compatible “trace” model

2nd Ed. AP210 Tools and Translators (InterCax / LKSoft)

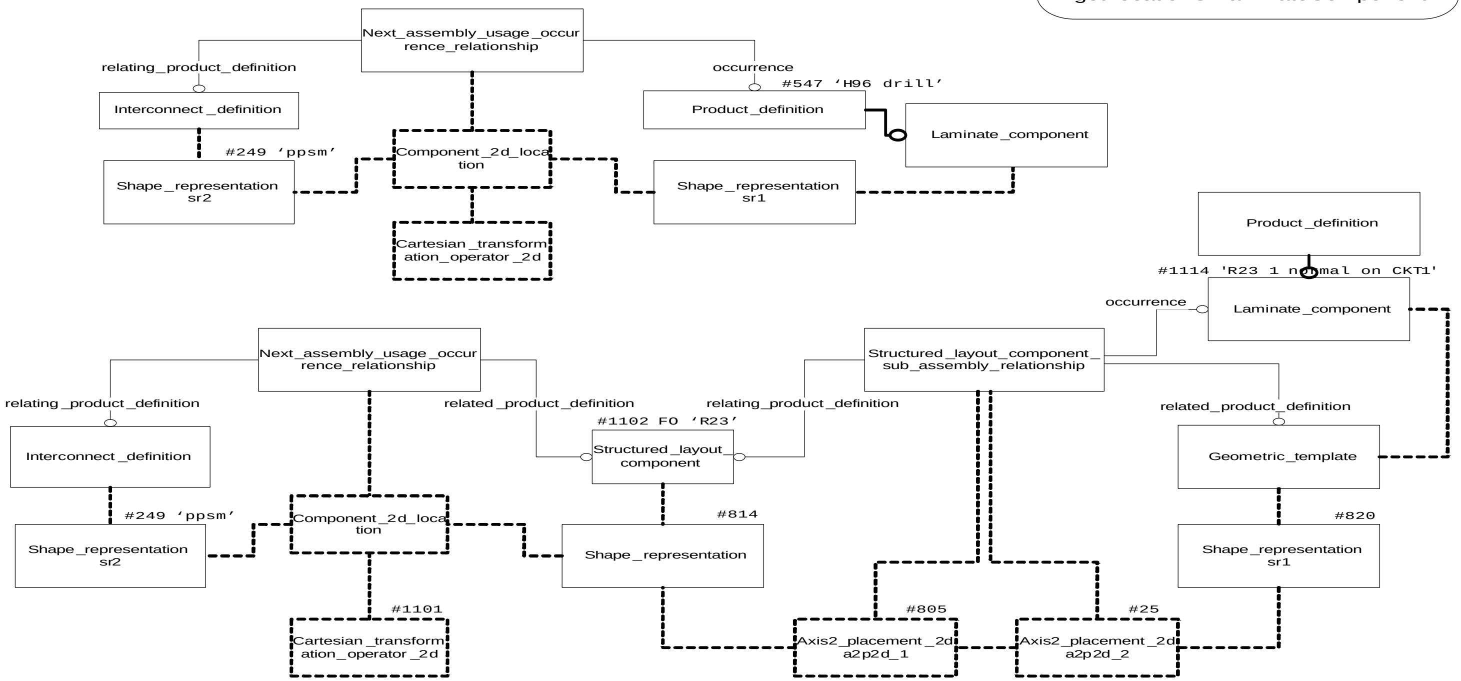
- IDA-STEP Product Family – viewers, converters, editors, diff tools
- JSDAI™ Toolkit - AP203, 210, 214, 236, 239, integration of ARM and AIM/MIM concepts

Translator	Xxx->AP210
Mentor / Boardstation	available
Mentor / PADS	available
Mentor / Expedition	available
Zuken - Visula / CR5000 / CADSTAR (CADIF)	available
Cadence / OrCAD	Q3'07
Cadence / Allegro	Q2'07
CadSoft / EAGLE	available

2^{nd} Ed. AP210 Recommended Practices - www.wikistep.org

- 2nd Edition of STEP AP-210 adopts the new modular architecture to promote interoperability with other STEP APs such as AP203 Ed. 2 and AP214.
- Translators from several common native ECAD representations for the 2nd Edition of AP-210 are available and/or under development.
- Breadth of the AP-210 standard, and the formidable learning curve associated with its adoption may challenge implementers.
- Currently working to define, document, and implement a series of procedures for the extraction of PCA and PCB data from a 2nd Edition AP-210 representation of an electronic assembly. (work funded by NIST)
  - Serve as a bridge for vendors looking to add AP210 import / export capabilities into existing tools as well as those developing design and/or manufacturing tools with native AP-210 capabilities.

getLocationOfLaminateComponent

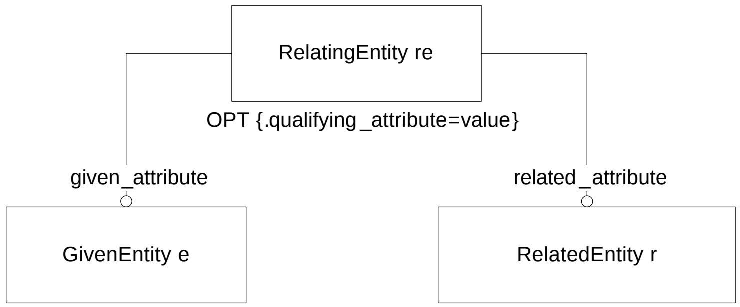



```
// Returns between 0 and 3 transformations that must be applied sequentially to locate the shape _representation of the laminate _component with respect to the shape _representation
// of the interconnect _definition (pcb). Query may be applied to either a Laminate _component that is part of a Structured _layout _component or a Laminate _component located directly on the Pcb .

[Cartesian_transformation _operator _2d; Axis2_placement_2d; Axis2_placement_2d] getLocationOfLaminateComponent (
        Interconnect_definition id , Laminate_component lc , Shape_representation sr 1, Shape_representation sr 2)
{
        Structured_layout_component_sub_assembly_relationship slcsar = referencingEntityOp (lc)
                where {slcsar.related_product_definition ->lc }

        If (slcsar !=null)
        {
                structured_layout_component slc = referencedEntityOp (slcsar)
                        where       {slcsar.relating_product_definition -> slc }

                structured_template st = referencedEntityOp (slc)
                        where       {slc.relating_product_definition -> st}

                shape_representation srOfslc = getShapeRepresentationOfProductDefinitionShape (st);

                [a2p2d1; a2p2d2] = getAxisPlacementOfSLCSAR (slcsar, sr1, srOfslc);

                next_assembly_occurrence_usage_relationship naour = referencingEntityOp (slc)
                        where {naour.related_product_definition ->slc }

                cartesian_transformation_operator_2d cto2d = getCartesianTransformationOfNAUOR (naour, srOfslc, sr2)

                return [cto2d; a2p2d1; a2p2d2]
        }
        else // Laminate_component is not part of a Structured _layout_component
        {
                next_assembly_occurrence_usage_relationship naour = referencingEntityOp (lc)
                        where       {naour.related_product_definition ->lc }
                                    {naour.relating_product_definition ->id }

                cartesian_transformation_operator_2d cto2d = getCartesianTransformationOfNAUOR (naour, sr1, sr2)

                return [cto2d; null; null]
        }
}
```

2nd Ed. Query Implementation

- ***Use of code generation (templates) to instantiate specific query operations.***
  - Automate “mechanical” portions of the JSDAI-specific code generation
  - Reduce implementation time and permit alternate backing implementations
    - Query trace / debugging code
    - Strict / loose operation checking (i.e. result uniqueness, qualifications)
    - Experiment with alternate query strategies (i.e. maps)

Example: Relationship Operation

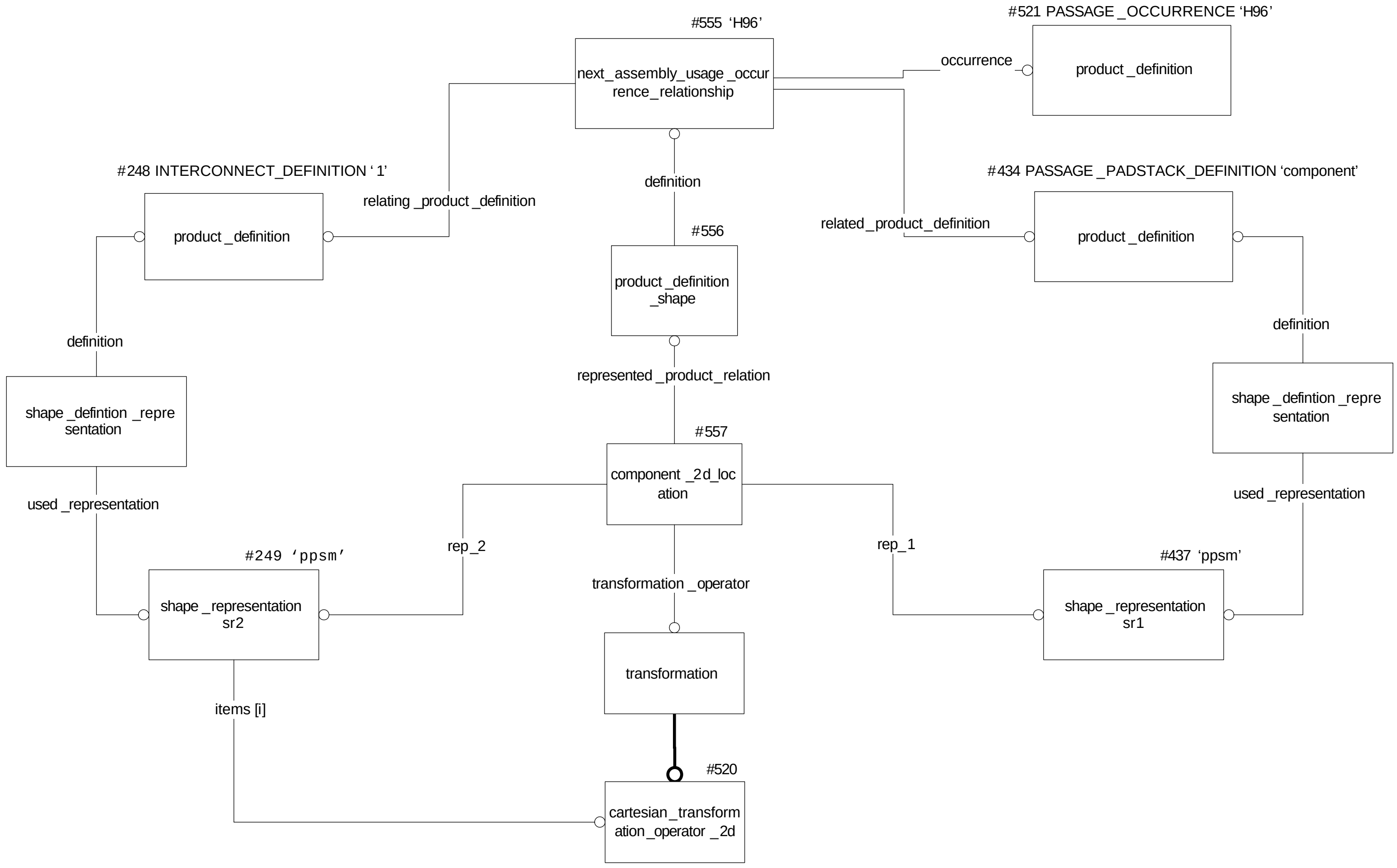



```
// Given: GivenEntity e
// Returns: RelatedEntity r
// Where r is related to e through RelatingEntity re
//   e<-re.given_attribute
//   re.related_attribute->r
//   OPT re.qualifying_attribute=value

// Returns an entity r of type RelatedEntity that is related to the given entity e of type GivenEntity through the a relating entitiy
// re of type RelatingEntity that references GivenEntity through the attribute given _attribute and RelatedEntity through the
// attribute related _attribute. Optionally, the relating entity may be qualified through the additional requirement that
// re.qualifying_attribute = value.
// If there is no entity that satisfies this requirement , returns null.

RelatedEntity r = relatedEntityOp(GivenEntity e)
    where {RelatingEntity re}
        {e <- re.given_attribute}
        {re.related_attribute -> r}
        OPT {re.qualifying_attribute=value}
{
    // implementation
    return RelatedEntity r … .
}
```

Sample Relationship Operation - sr_relatedTo_pd_through_sdr

```
/**
 * Given a product_definition,
 * this method will return a shape_representation
 * related by a shape_definition_representation.
 * <p>
 * product_definition <-
 * shape_definition_representation.definition
 * shape_definition_representation
 * {shape_definition_representation
 * shape_definition_representation.used_representation ->
 * shape_representation =>
 * <p>
 * @param e1 the starting entity of type product_definition
 */

public EShape_representation sr_relatedTo_pd_through_sdr (EProduct_definition e1) throws SdaiException
{
        AShape_definition_representation a_relationship = new AShape_definition_representation();
        CShape_definition_representation.usedinDefinition(null, (EProduct_definition)e1, null, a_relationship);
        SdaiIterator it_entities = a_relationship.createIterator();
        while (it_entities.next())
        {
                EShape_definition_representation e_relationship = (EShape_definition_representation) a_relationship.getCurrentMemberEntity(it_entities);
                        {
                                //System.out.println("found relationship: " + e_relationship.getPersistentLabel());
                                EShape_representation e2 = (EShape_representation) e_relationship.getUsed_representation(null);
                                //System.out.println("used_representation: " + e2.getPersistentLabel());
                                return e2;
                        }
        }
        return null;
}
```

Sample Query – Get Transformation for NAUOR



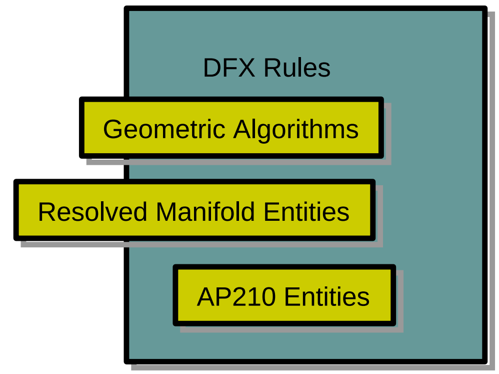
Sample Query - Get Transformation for NAUOR

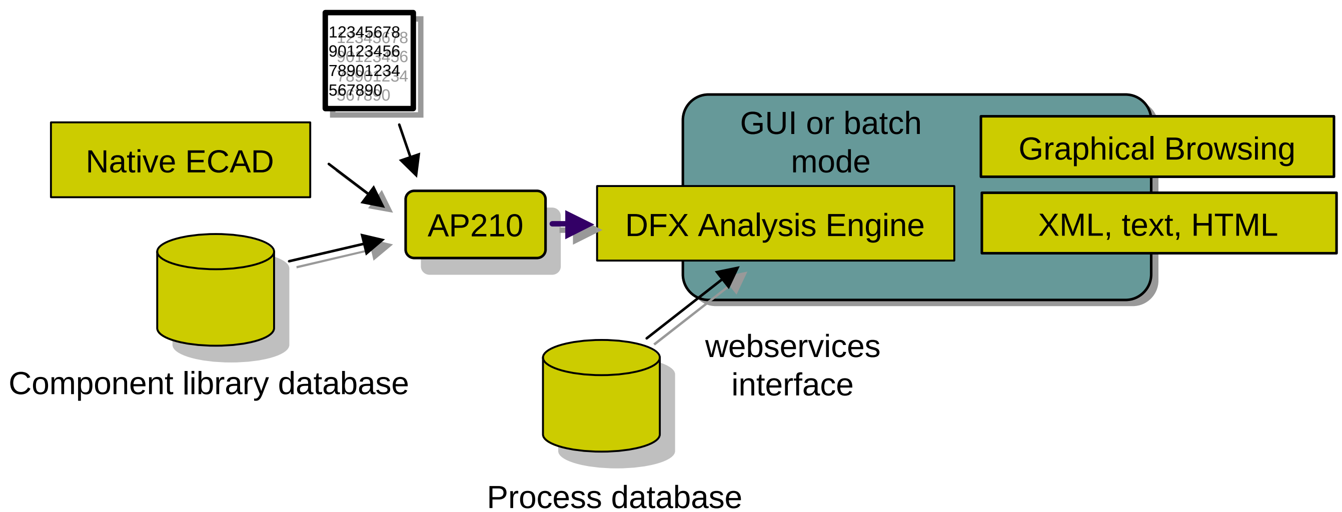
// Returns a cartesian_transformation_operator_2d in the case that a relating component_2d_location exists
// for the given next_assembly_usage_occurrence_relationship. The c2dl is qualified by the two given shape_representations.
// If no such transformation exists, the query returns null.

```
public ECartesian_transformation_operator_2d getCartesianTransformationOfNAUOR(
        ENext_assembly_usage_occurrence_relationship nauor,
        EShape_representation sr1,
        EShape_representation sr2) throws SdaiException
{
    EProduct_definition_shape pds = ops.pds_referencing_nauor(nauor);

    if (pds == null)
        return null;

    AComponent_2d_location a_c2dl = ops.All_c2dl_referencing_pds(pds);

    EComponent_2d_location e_c2dl = ops.c2dl_referencingGiven(a_c2dl, sr1, sr2);

    if (e_c2dl != null)
    {
        ECartesian_transformation_operator_2d cto2d = ops.cto2d_referencedBy_c2dl(e_c2dl);
        return cto2d;
    }
    return null;
}
```

DFx Model and Process Flow

- Internal data structure respects key AP210 concepts such as padstacks, stratum features, layer connection points, physical networks
- Comprehensive PCB API that allows efficient traversal of model for rule processing.
  - Graph-based connectivity algorithms
- Dual geometric representation
  - Comprehensive geometric library
  - Native AP210 / ECAD design entities
  - Resolved geometry for physical layers
  - Mapping between AP210 entities and resolved geometric entities



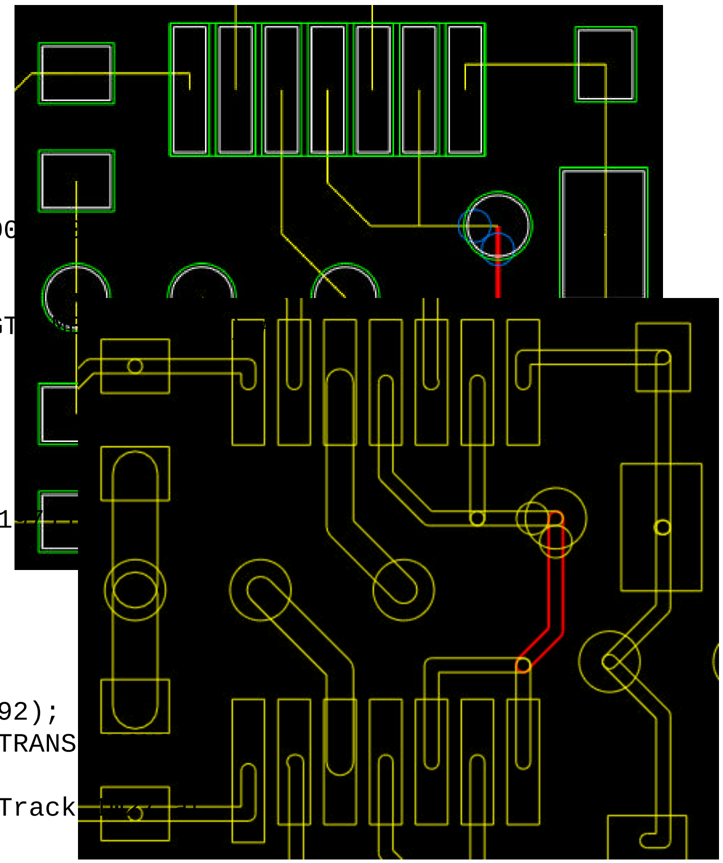


Preprocessing of AP210 entities to geometric regions

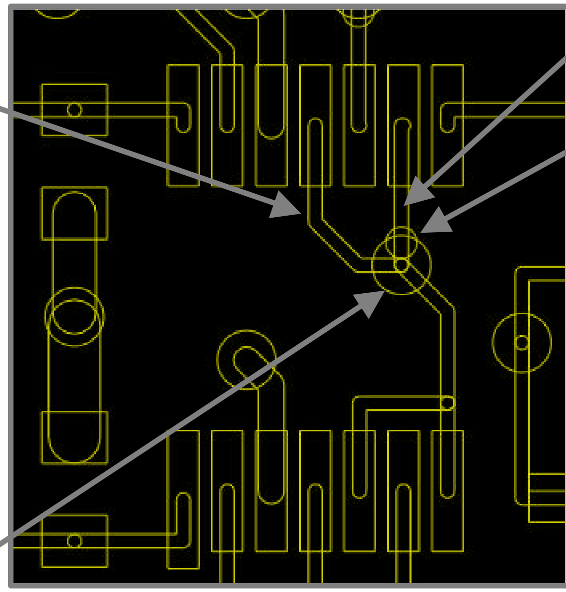
- A trace is defined by a centerline, width, and transition and end conditions in AP210

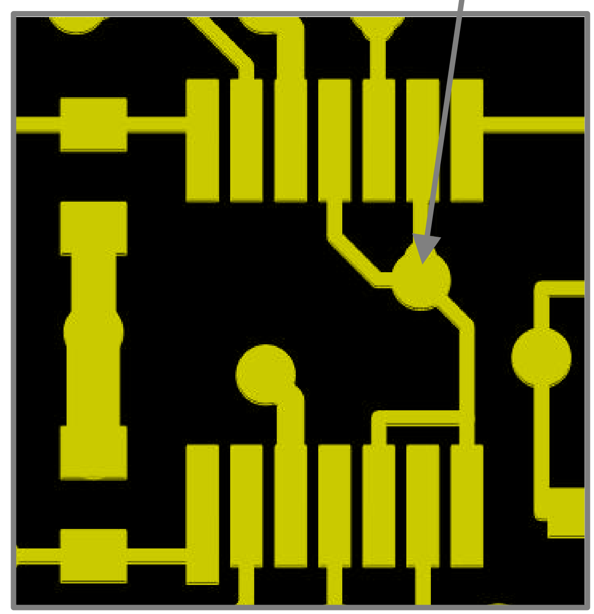
```
#109942=LENGTH_MEASURE_WITH_UNIT(LENGTH_MEASURE(0.20320000
#109943=CURVE_STYLE('Curve style
2',.ROUND.,#109942,$,.EXTEND.,.ROUND.,#109944);
#109944=(LENGTH_MEASURE_WITH_UNIT()MEASURE_WITH_UNIT(LENGT
28) UNCERTAINTY_MEASURE_WITH_UNIT(' ',$));
#109945=TRACE_TEMPLATE('NULL','Trace template
0.20320000000000002',$,$,$,#109941,#109943);

#133191=COMPOSITE_CURVE('centreline',(#133198),.F.);
#133192=POLYLINE('',(#133193,#133194,#133195,#133196,#1331
#13
#13
#13
#13
#13
#13                                                   192);
#13        _            _        _     _    _          _TRANS
109945,#34459,#105284,#133191);
#133200=NEXT_HIGHER_ASSEMBLY_RELATIONSHIP(#23,#133199,$,'Track
L1',$);
```

- For geometric processing, this is converted to a resolved manifold entity (polygon).



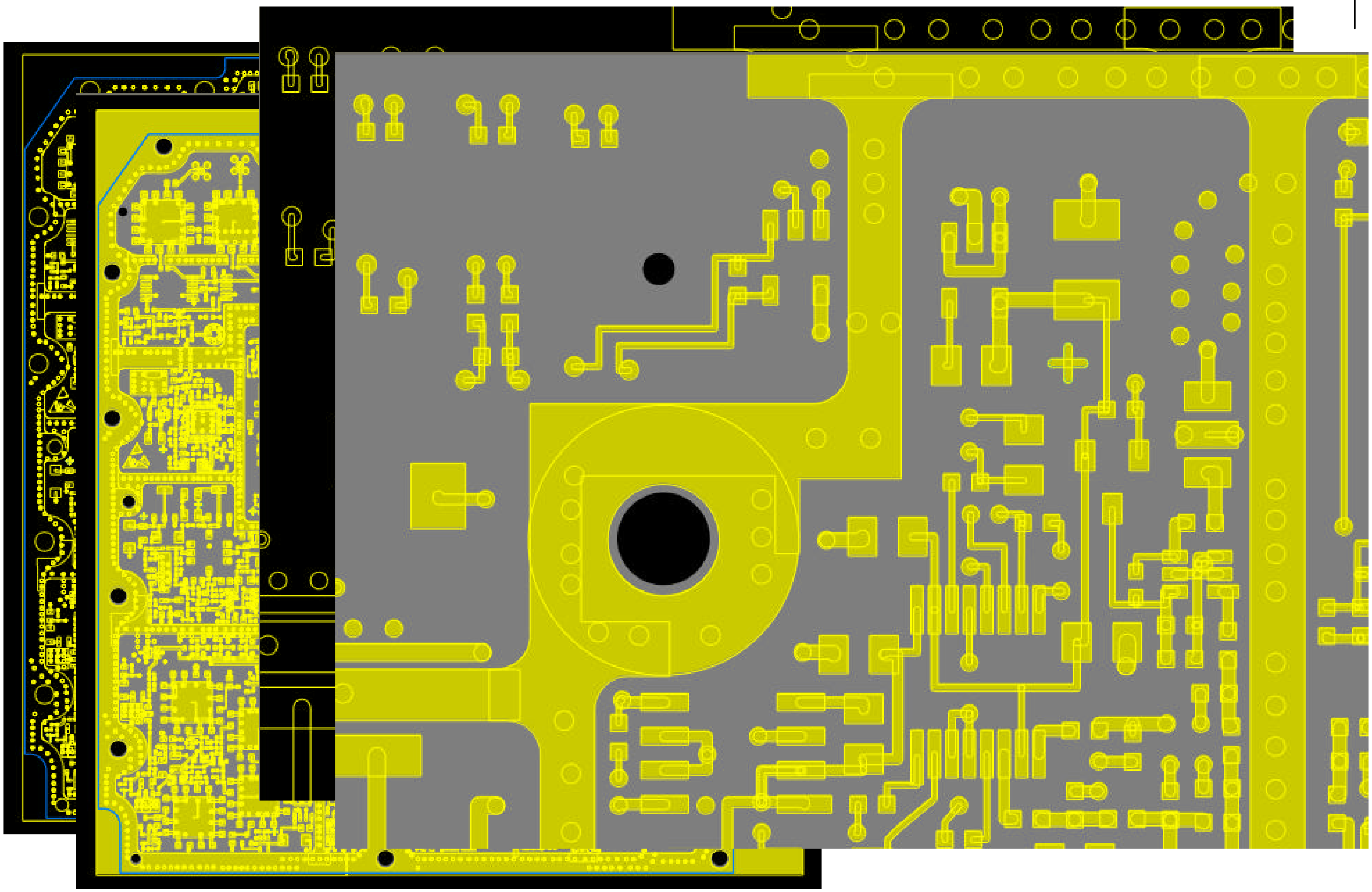
Mapping AP210 entities to resolved geometry - copper





- Overlapping polygons are resolved in a secondary processing step.

Mapping AP210 entities to resolved geometry - copper

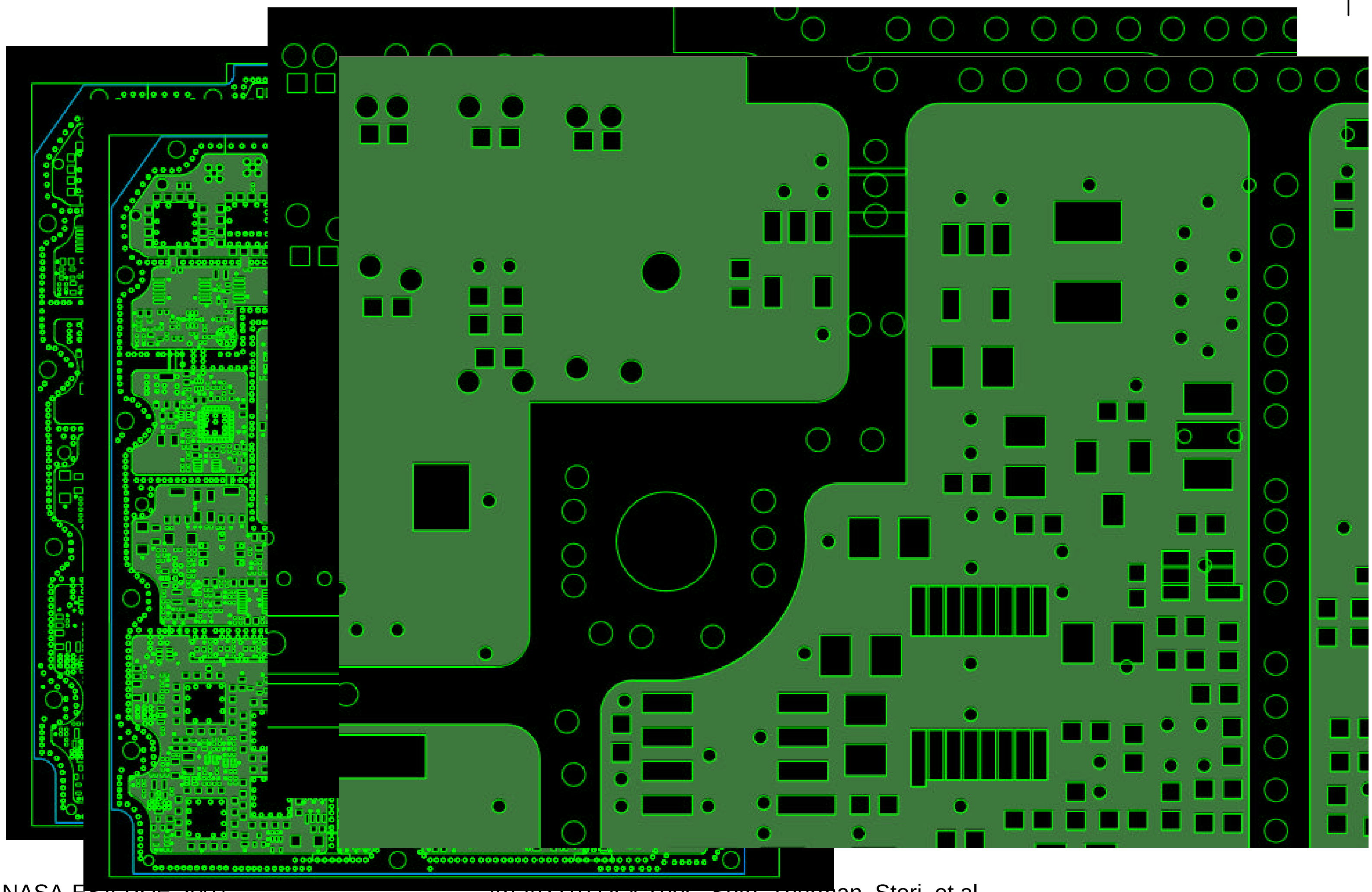


Mapping AP210 entities to resolved geometry - SM

- a single SM cutout can overlap hundreds of other entities!

AF49 cutout 1 for base for negative layer S/RESIST T

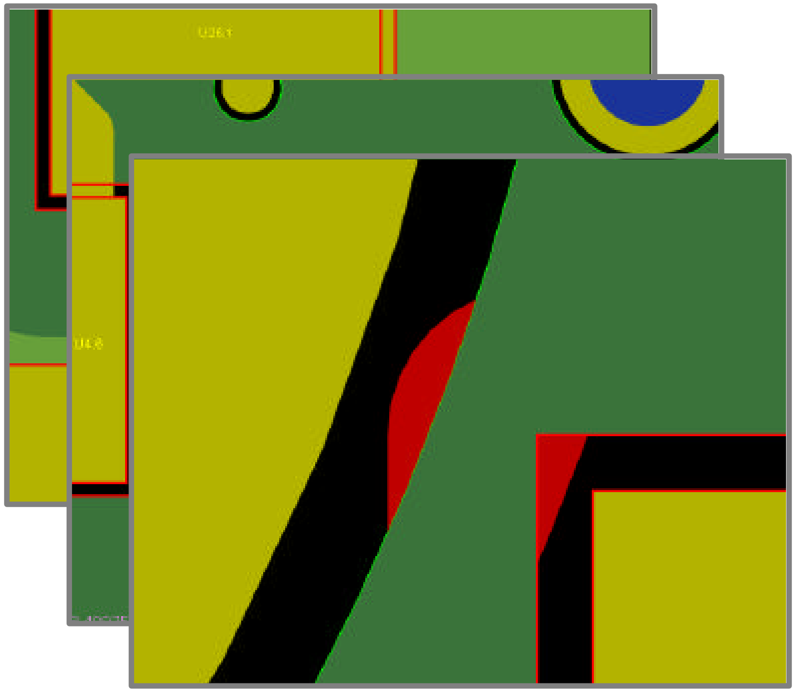
Mapping AP210 entities to resolved geometry - SM

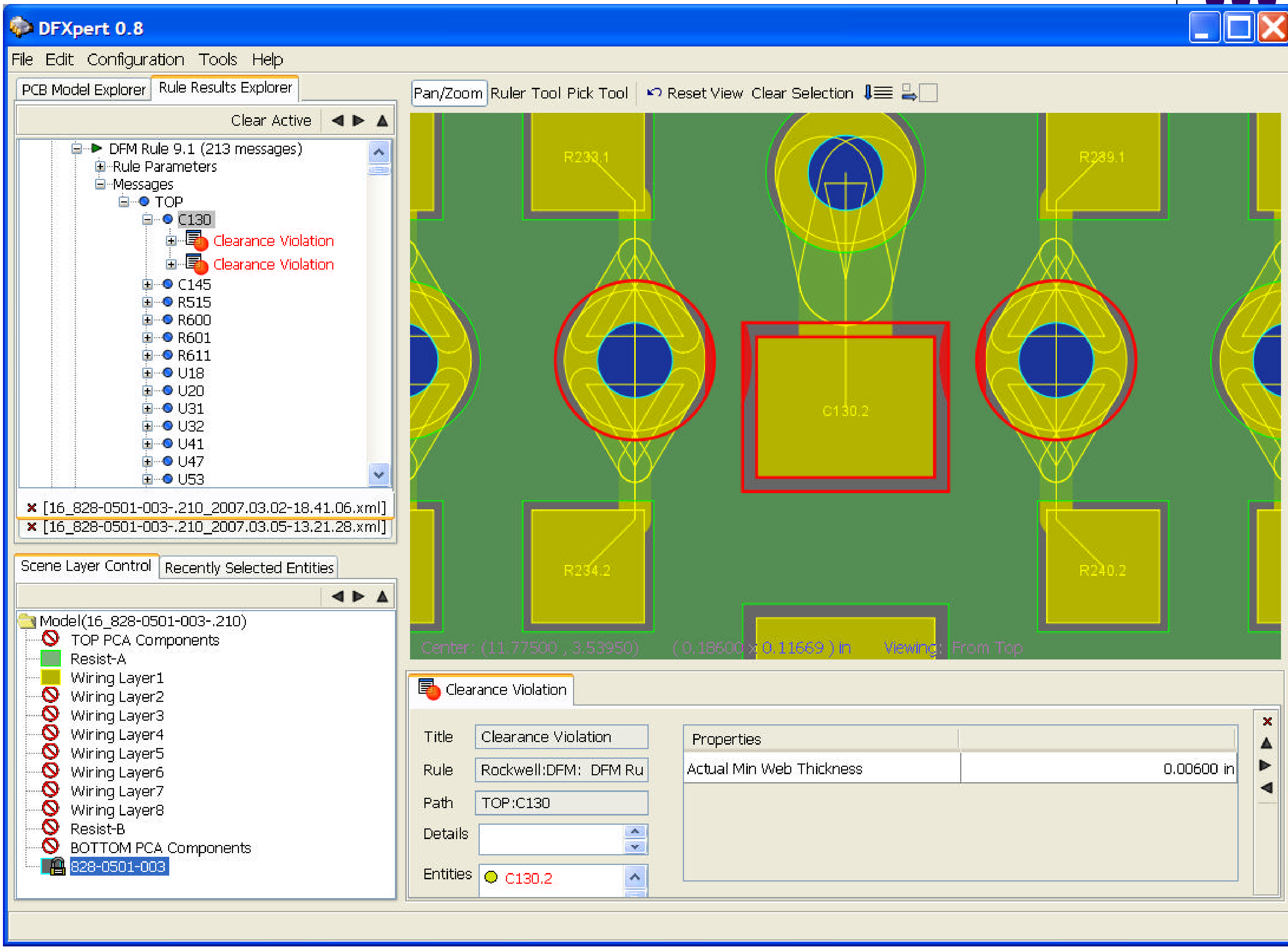


DFM example: Minimum SM web thickness

Minimum solder mask web must be no less than “x”

- Resolve intersecting and overlapping positive and negative soldermask boundaries.
- Violations limited to thin webs in proximity to attach pads.
- Direct visualization of violating geometry. Violations mapped back to design entities for reporting.

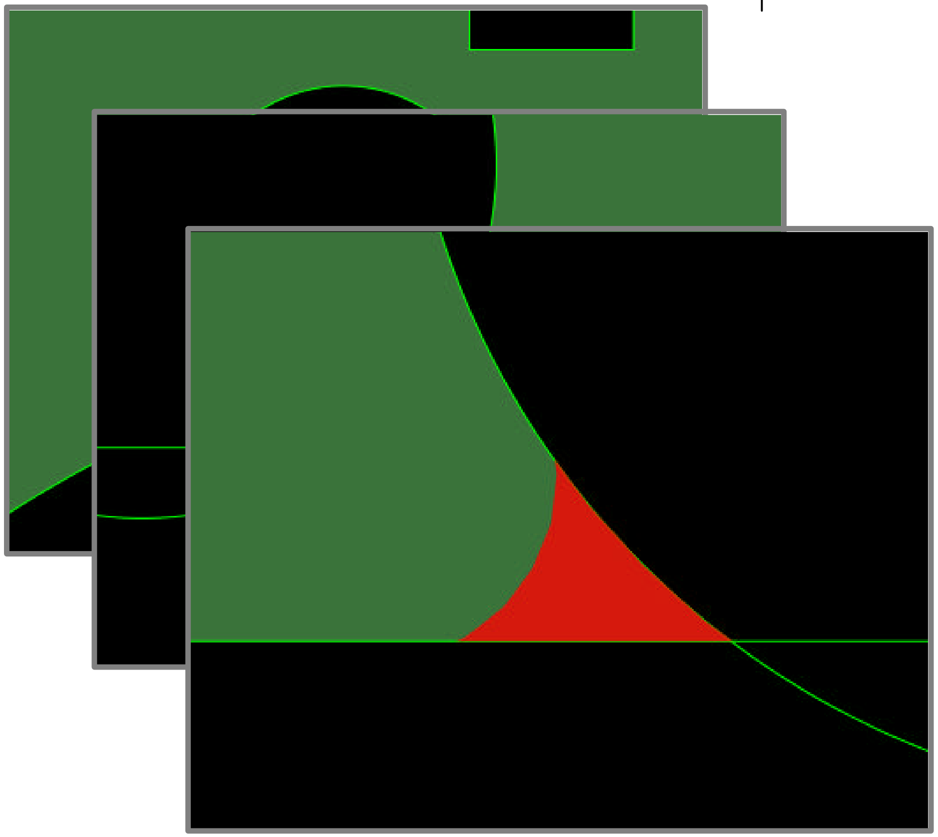


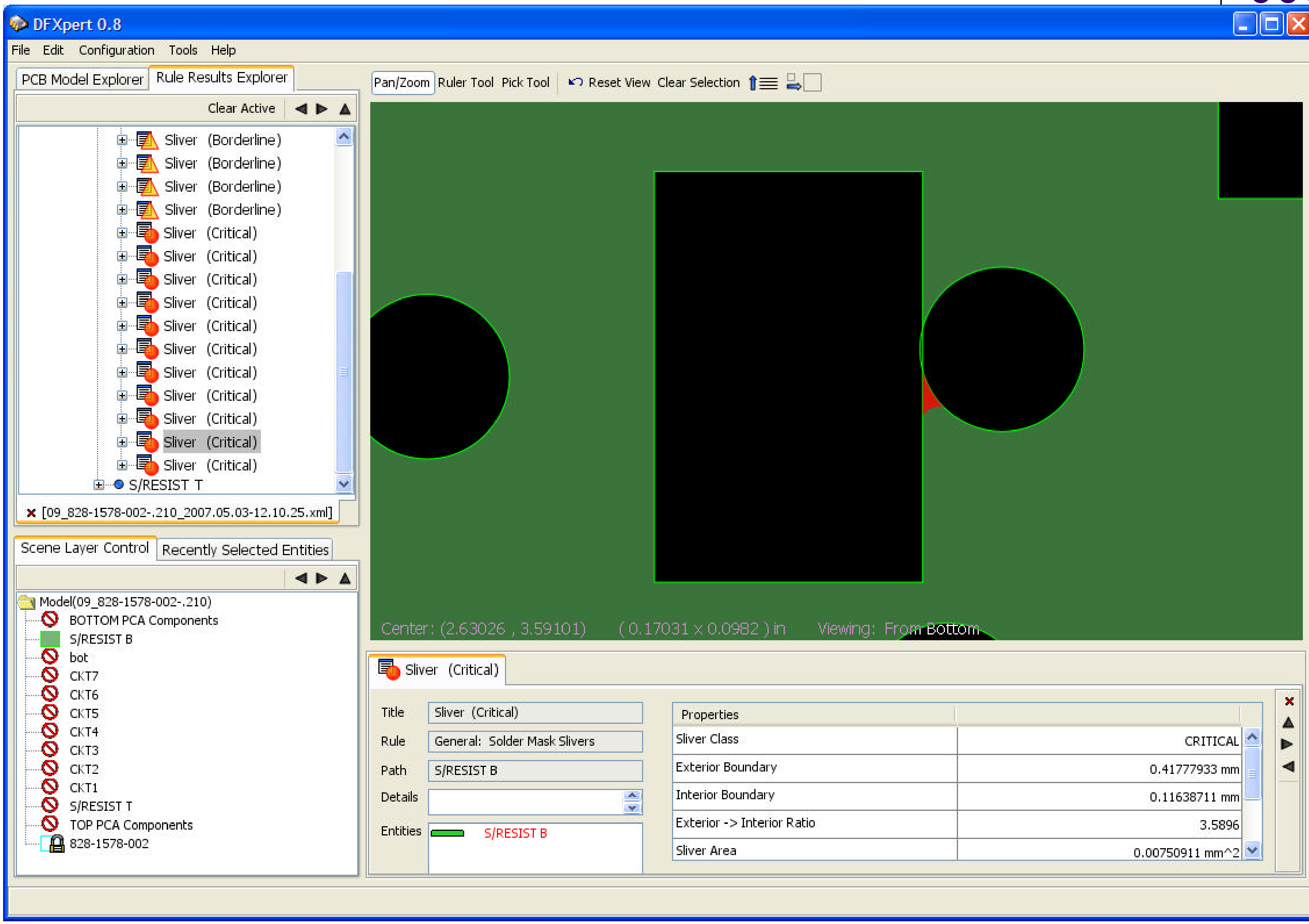

DFXpert 0.8
File Edit Configuration Tools Help
PCB Model Explorer
Rule Results Explorer
Clear Active
DFM Rule 9.1 (213 messages)
Rule Parameters
Messages
TOP
C130
Clearance Violation
Clearance Violation
C145
R515
R600
R601
R611
U18
U20
U31
U32
U41
U47
U53
[16_828-0501-003-.210_2007.03.02-18.41.06.xml]
[16_828-0501-003-.210_2007.03.05-13.21.28.xml]
Scene Layer Control
Recently Selected Entities
Model(16_828-0501-003-.210)
TOP PCA Components
Resist-A
Wiring Layer1
Wiring Layer2
Wiring Layer3
Wiring Layer4
Wiring Layer5
Wiring Layer6
Wiring Layer7
Wiring Layer8
Resist-B
BOTTOM PCA Components
828-0501-003
Pan/Zoom
Ruler Tool
Pick Tool
Reset View
Clear Selection
R233.1
R239.1
C130.2
R234.2
R240.2
Center: (11.77500 , 3.53950)
(0.18600 x 0.11669) in
Viewing: From Top
Clearance Violation
Title
Clearance Violation
Rule
Rockwell:DFM: DFM Ru
Path
TOP:C130
Details
Entities
C130.2
Properties
Actual Min Web Thickness
0.00600 in

DFM example: Soldermask Slivers

Identify soldermask features that could cause quality issues due to delamination.

- Offsetting and boolean operations on soldermask geometry
- Classification of candidate slivers based on boundary analysis.
- Direct visualization of violating geometry.

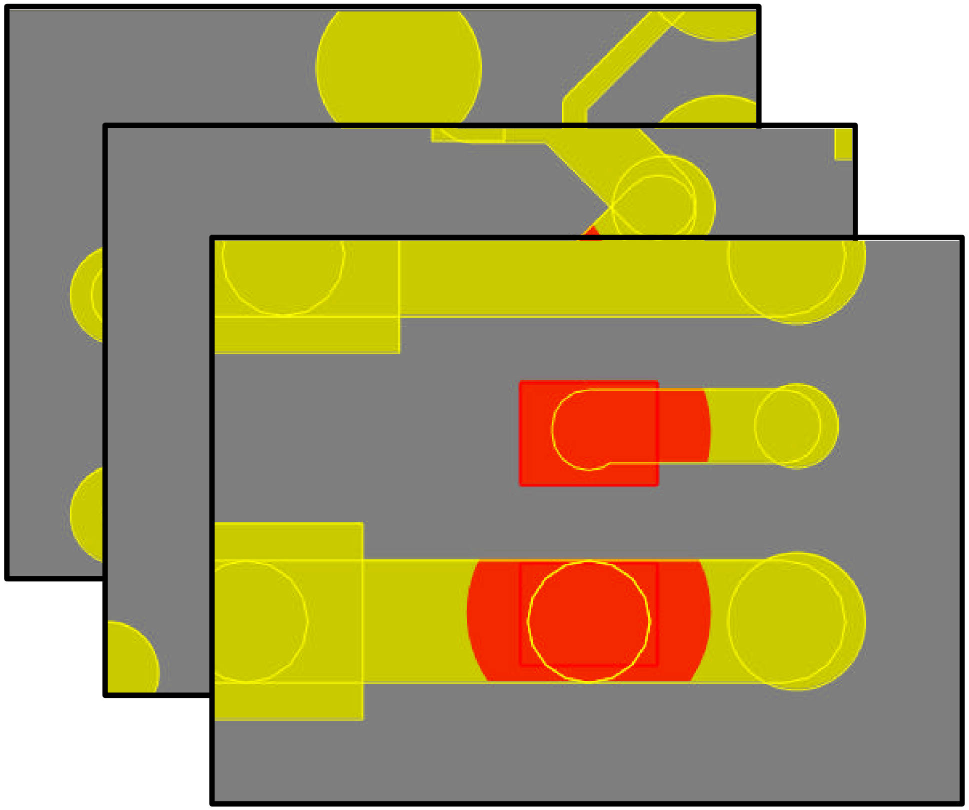


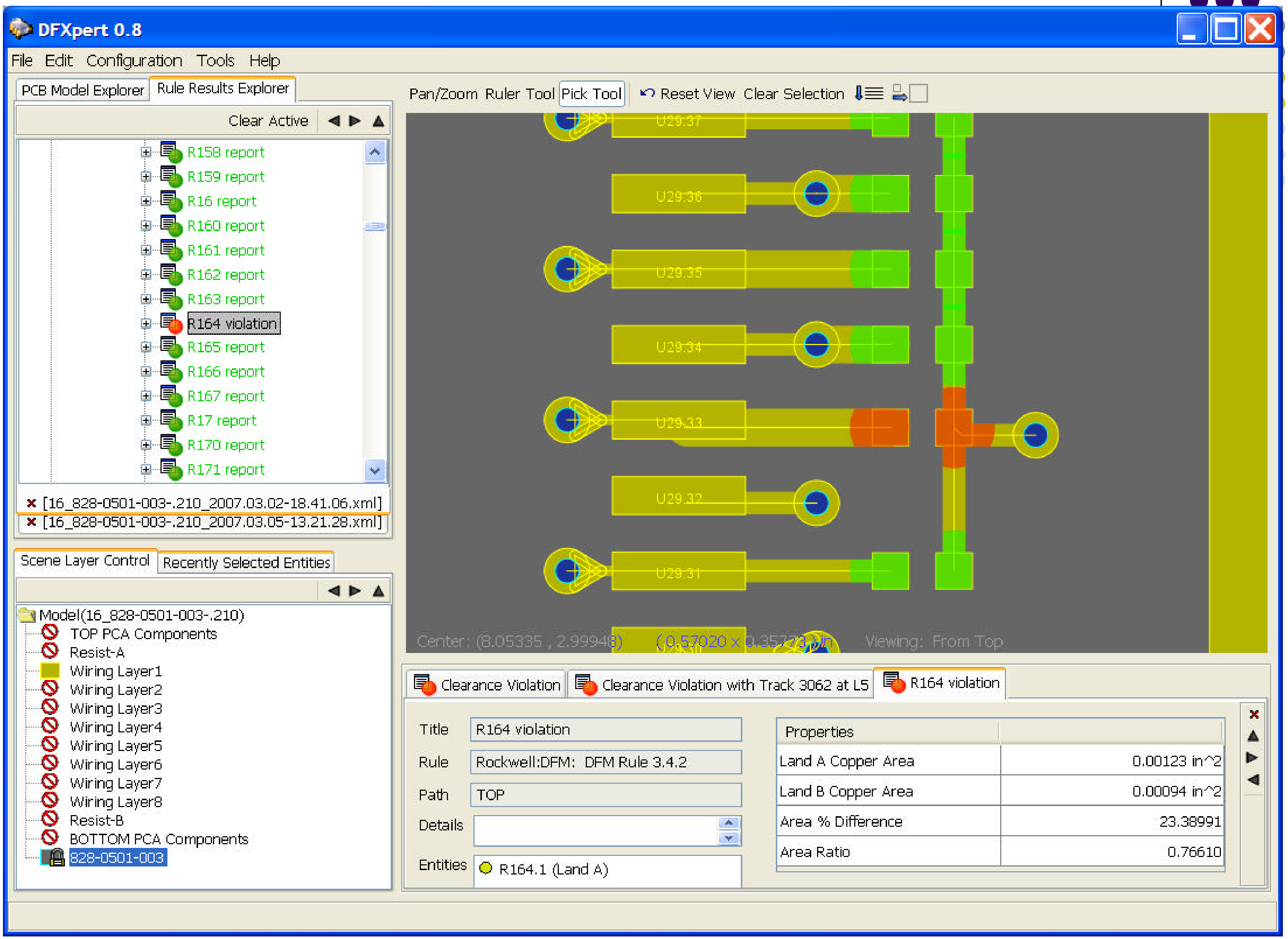


DFXpert 0.8
File Edit Configuration Tools Help
PCB Model Explorer
Rule Results Explorer
Clear Active
Sliver (Borderline)
Sliver (Borderline)
Sliver (Borderline)
Sliver (Borderline)
Sliver (Critical)
Sliver (Critical)
Sliver (Critical)
Sliver (Critical)
Sliver (Critical)
Sliver (Critical)
Sliver (Critical)
Sliver (Critical)
Sliver (Critical)
Sliver (Critical)
Sliver (Critical)
S/RESIST T
[09_828-1578-002-.210_2007.05.03-12.10.25.xml]
Scene Layer Control
Recently Selected Entities
Model(09_828-1578-002-.210)
BOTTOM PCA Components
S/RESIST B
bot
CKT7
CKT6
CKT5
CKT4
CKT3
CKT2
CKT1
S/RESIST T
TOP PCA Components
828-1578-002
Pan/Zoom
Ruler Tool
Pick Tool
Reset View
Clear Selection
Center: (2.63026 , 3.59101)
(0.17031 x 0.0982) in
Viewing: From Bottom
Sliver (Critical)
Title
Sliver (Critical)
Rule
General: Solder Mask Slivers
Path
S/RESIST B
Details
Entities
S/RESIST B
Properties
Sliver Class
CRITICAL
Exterior Boundary
0.41777933 mm
Interior Boundary
0.11638711 mm
Exterior -> Interior Ratio
3.5896
Sliver Area
0.00750911 mm^2

DFA example: Copper Balance

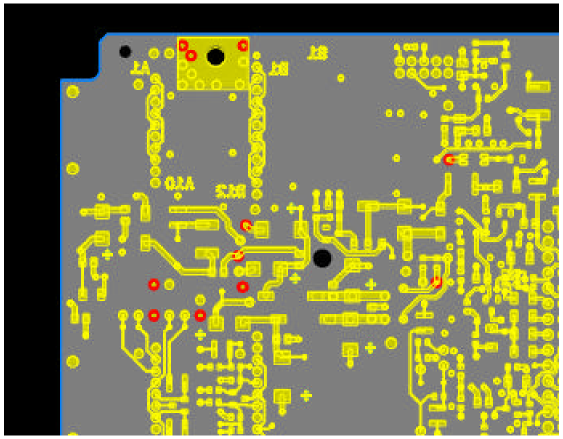
- Analyze copper balance in proximity of lands associated with certain PCA components
  - Requires geometric resolution of full copper connectivity
  - Computation of integral properties (i.e. area)
  - Quantitative reporting
  - Direct visual feedback

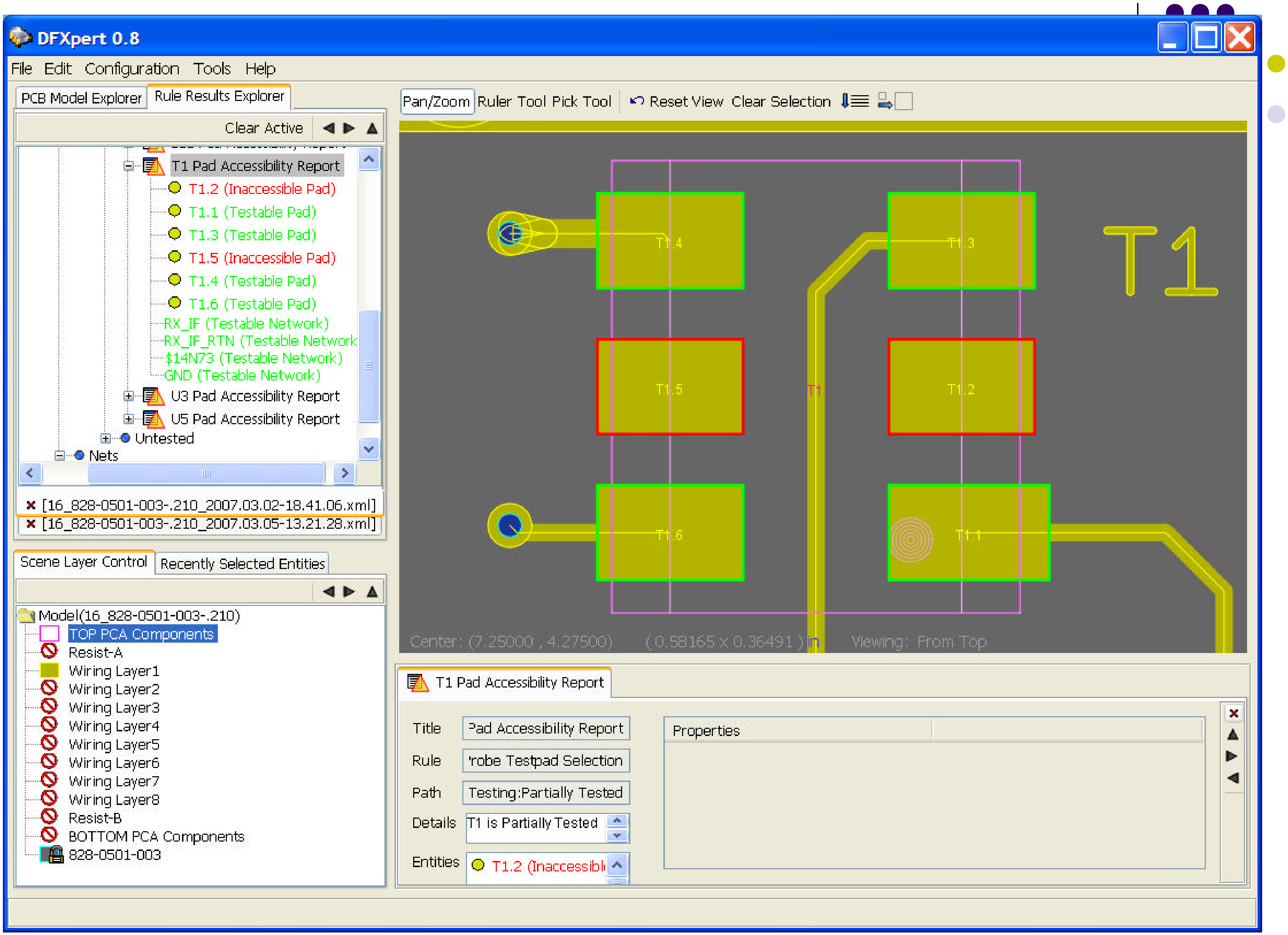



DFXpert 0.8
File Edit Configuration Tools Help
PCB Model Explorer
Rule Results Explorer
Clear Active
R158 report
R159 report
R16 report
R160 report
R161 report
R162 report
R163 report
R164 violation
R165 report
R166 report
R167 report
R17 report
R170 report
R171 report
[16_828-0501-003-.210_2007.03.02-18.41.06.xml]
[16_828-0501-003-.210_2007.03.05-13.21.28.xml]
Scene Layer Control
Recently Selected Entities
Model(16_828-0501-003-.210)
TOP PCA Components
Resist-A
Wiring Layer1
Wiring Layer2
Wiring Layer3
Wiring Layer4
Wiring Layer5
Wiring Layer6
Wiring Layer7
Wiring Layer8
Resist-B
BOTTOM PCA Components
828-0501-003
Pan/Zoom Ruler Tool Pick Tool
Reset View Clear Selection
U29.37
U29.36
U29.35
U29.34
U29.33
U29.32
U29.31
Center: (8.05335 , 2.99948)
Viewing: From Top
Clearance Violation
Clearance Violation with Track 3062 at L5
R164 violation
Title R164 violation
Rule Rockwell:DFM: DFM Rule 3.4.2
Path TOP
Details
Entities R164.1 (Land A)
Properties
Land A Copper Area 0.00123 in^2
Land B Copper Area 0.00094 in^2
Area % Difference 23.38991
Area Ratio 0.76610

DFT example: Testpad Identification

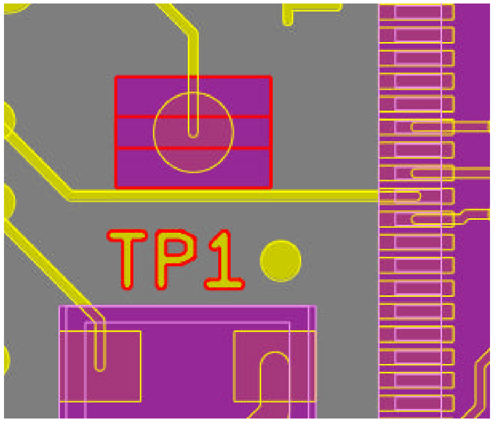
- Support requirements for multiple test processes (FP, ICT)
- Rules combine
  - Netlist – pad connectivity
  - PCA clearance
  - PCB clearance, proximity
  - Text strings in close proximity
- Requirements
  - Search for a series of test pad candidates that meet certain criteria
    - Sufficient size for testing equipment
    - Satisfies clearance and accessibility requirements
  - Filter, summarize, and report by component and net.

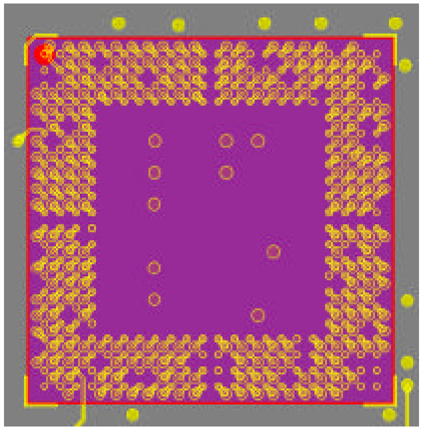



DFXpert 0.8
File Edit Configuration Tools Help
PCB Model Explorer
Rule Results Explorer
Clear Active
T1 Pad Accessibility Report
T1.2 (Inaccessible Pad)
T1.1 (Testable Pad)
T1.3 (Testable Pad)
T1.5 (Inaccessible Pad)
T1.4 (Testable Pad)
T1.6 (Testable Pad)
RX_IF (Testable Network)
RX_IF_RTN (Testable Network
$14N73 (Testable Network)
GND (Testable Network)
U3 Pad Accessibility Report
U5 Pad Accessibility Report
Untested
Nets
[16_828-0501-003-.210_2007.03.02-18.41.06.xml]
[16_828-0501-003-.210_2007.03.05-13.21.28.xml]
Scene Layer Control
Recently Selected Entities
Model(16_828-0501-003-.210)
TOP PCA Components
Resist-A
Wiring Layer1
Wiring Layer2
Wiring Layer3
Wiring Layer4
Wiring Layer5
Wiring Layer6
Wiring Layer7
Wiring Layer8
Resist-B
BOTTOM PCA Components
828-0501-003
Pan/Zoom
Ruler Tool
Pick Tool
Reset View
Clear Selection
T1
T1.4
T1.3
T1.5
T1.2
T1.6
T1.1
Center: (7.25000 , 4.27500)
(0.58165 x 0.36491) in
Viewing: From Top
T1 Pad Accessibility Report
Title
Pad Accessibility Report
Rule
Probe Testpad Selection
Path
Testing:Partially Tested
Details
T1 is Partially Tested
Entities
T1.2 (Inaccessibl
Properties

DFT example:
Support for Visual Inspection

- Require laminate text strings in close proximity to certain PCA components and testpoints
- Polarized components have consistent orientation <or> orientation symbols
- BGAs have visible orientation markings
  - Not obscured by component footprint
  - Must be in close proximity
  - Chamfer to indicant pin one
    - Shape recognition algorithm for identification





Case Study – RCI Designs

- 6 production designs from CR-5000 -> AP210
- part 21 physical files up to 35 MB (outer layers only)
- 48 DFM rules / 61 DFT rules
- Processing time
  (times below will be dramatically reduced in new implementation)
  - <10 min for 5 of 6.
  - ~1 hr for the 6th
- Number of design entities (on outer layers) up to:
  - 9,700 lands, 3,800 traces, 7,300 non-land copper regions
  - 56,000 polygons
  - 4,200,000 vertices
  - <8 min processing time

Status

- AP210 has been vetted in a challenging production environment
  - Comprehensive AP210 model supports a wide range of DFA / DFM / DFT analysis
  - Translators from major ECAD formats are available
- DFx capabilities have been developed and demonstrated
  - Robust geometric library integrated with PCB model
  - Flexible architecture to support rapid customization and rule prototyping
  - Results validated on hundreds of production designs by producibility and test engineers
  - Commercial implementation (DFXpert) complete pending 2nd Ed. update and regression testing