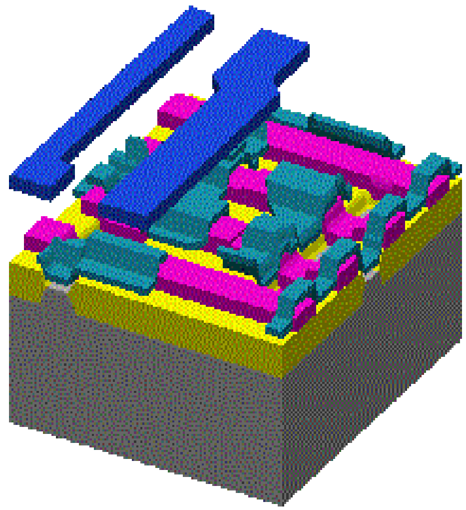
Standard 2D and 3D TCAD Inteface Issues



Robert W. Dutton and **Edwin C. Kan**

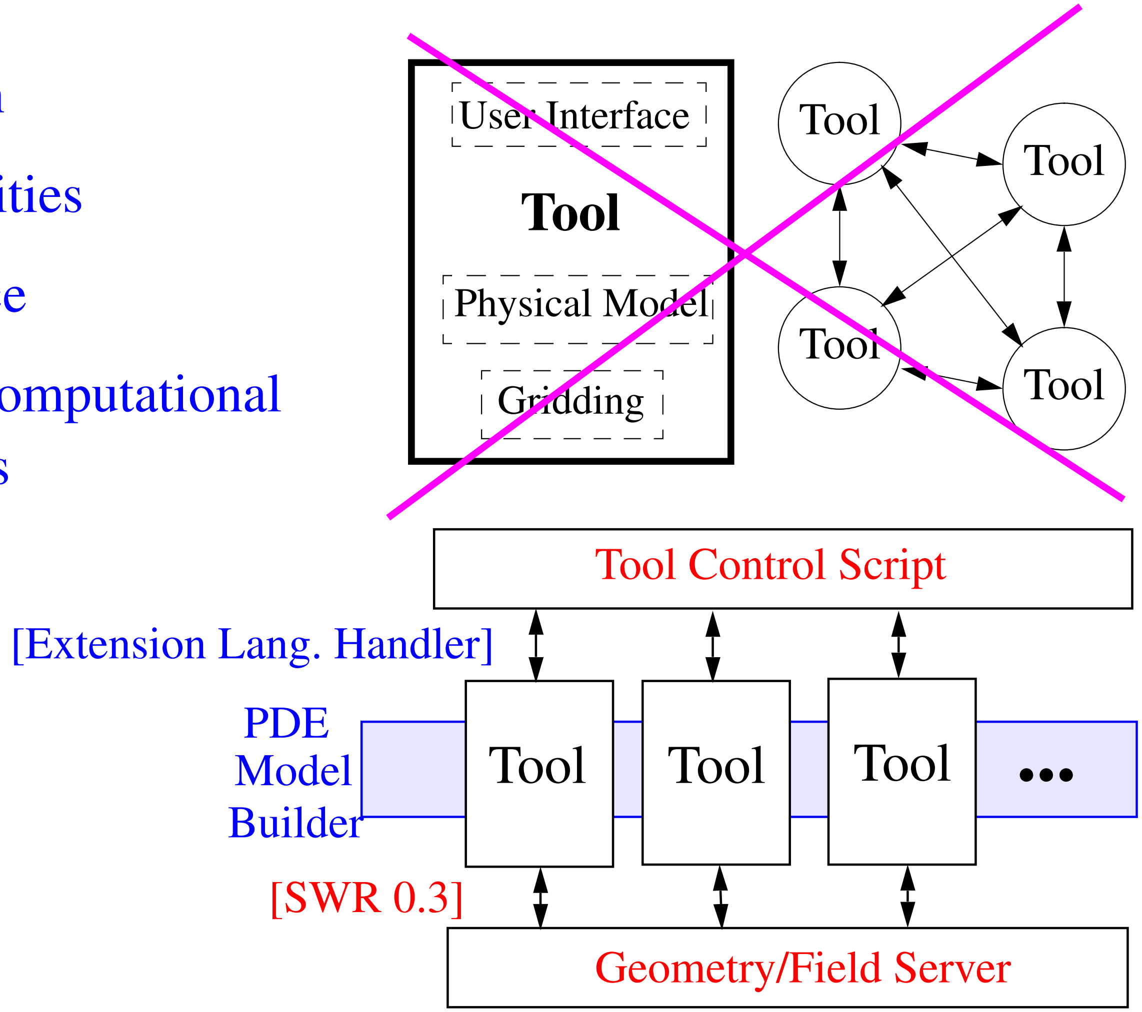
Stanford University

Outline

- Needs for TCAD tool integration
- History on TCAD tool integration (Stanford version)
- Present strategy: minimal SWR 0.3
  - common functionalities
  - extended functionalities
  - unified geometry, hybrid grid
  - a minimal API
- Tool examples using SWR 0.3
- Current status
- Conclusion

Needs for TCAD Tool Integration

- tool communication
- common functionalities
- unified user interface
- extension to other computational prototyping systems



History of TCAD Tool Integration

(Stanford Version)

- Interchange file format: no functionality provided
  - Sequential Access (ex: SUPREM structure file)
  - Hierarchical Access (ex: HDF-Vset)
- SWR 1.0 (Semiconductor Wafer Representation)
  - 2D, flexible access, limited functionalities
  - full-blown procedural interface for each class (total of ~700 public functions)
- SWR 2.0
  - 1-2-3D, flexible access, limited functionalities
  - procedural interface exists, no consensus reached, never implemented

Current Strategy for TCAD Tool Integration

(Stanford Version)

SWR 0.3: minimal procedural interface around geometry/field libraries that can provide essential functionality

- (total of ~140 access functions)
- No constraint on geometry/field data structure
- Major difference from file interchange format
  - Provide essential functionalities for maintenance and development
  - Still need persistent storage format, but only PI is specified
- Major difference from possible SWR 3.0
  - limited access enables easier re-use of established geometry/field libraries
  - unified geometry; hybrid grids

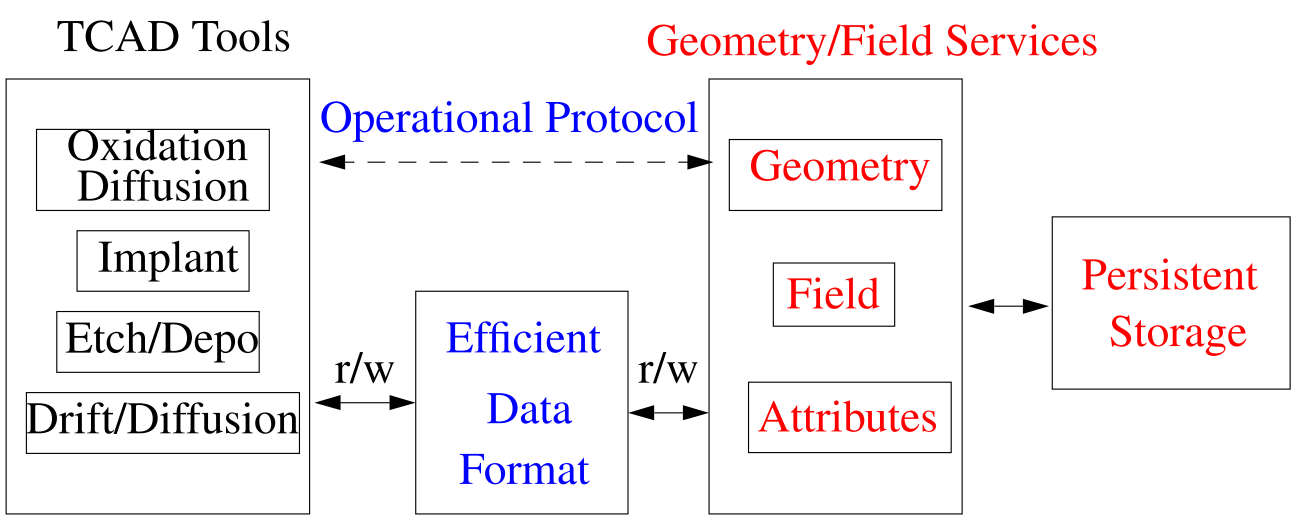
Geometry/Field Services

Common functionalities

- consistent field storage with flexible access
- error adaptation (element and nodal fields)
- boundary movement (geometrical and nodal)
- boundary smoothing (by curvature)
- consistent communication through geometry
- object validation

Extended applications

- interface to solid modeler (such as VIP-3D)
- link between 1-2-3D
- analytical geometry and fields



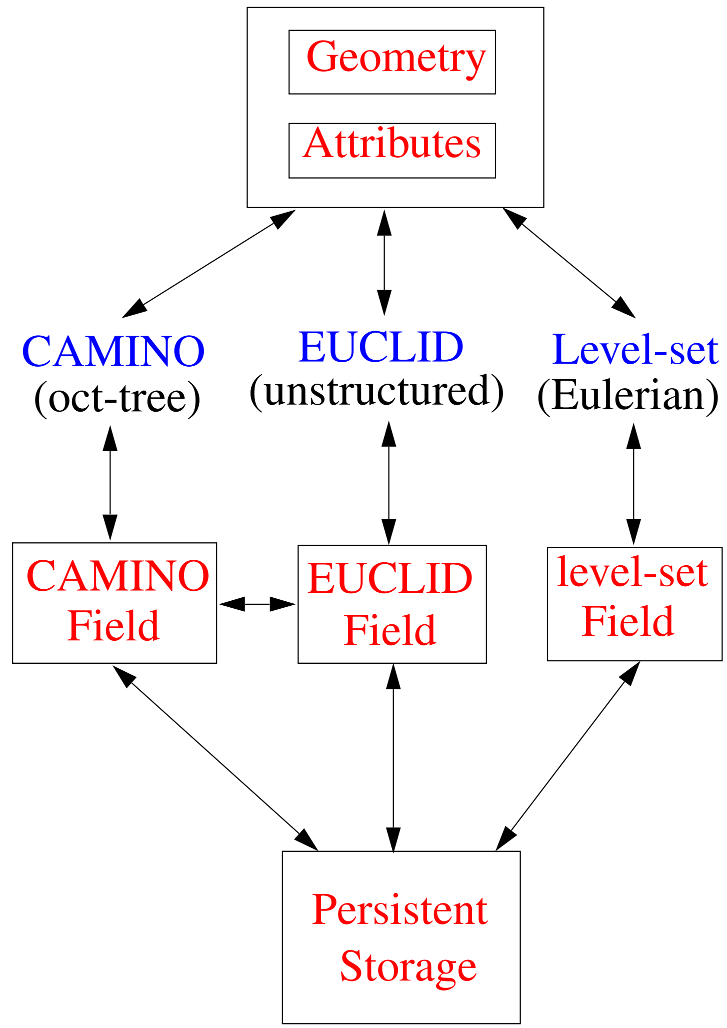
Operational Protocols for Geometry/Field Services

- Object-Oriented Design
  - abstraction: clearly defined objects (volume, surface, gridPoint, solution)
  - encapsulation: specification of methods, not data structure
  - ~~polymorphism: function overloading: reuse functions~~
  - ~~inheritance: object hierarchy: reuse data structure~~
- Unified boundary representation of geometry
  - hierarchical description: geometry; volume; surface; edge; point
  - consistency with grids by request only
- Hybrid implementation of gridders
  - different system of equations posts different requirements
  - no 3-D gridders universally good for complex geometry

Hybrid Implementation of Geometry/Field Services

(Stanford current version)

- CAMINO (oct-tree based)
  - initial gridding
  - robust adaptation
  - simpler element constraints for FE
- EUCLID (unstructured tetra.)
  - complex geometry
  - flexible adaptation
- Level-set (nonconforming Eulerian)
  - accurate curvature
  - accurate and robust boundary movement
- Open interface for other gridders



SWR 0.3 PI —— An Illustrative Glimpse

- **Geometry procedural interface**
  - construction and destruction: `swrGeometryAddVolume(swrGeometry *, swrVolume *)`
  - queries: `swrForEachVolumeinGeometry(swrGeometry *, swrVolume)`
  - attributes: `swrSetAttribute(swrGeometry *, char* attrName, char* attrValue)`
  - functionality: `levelSetGeometrySmooth(swrGeometry *, double curvatureThreshold)`
- **Volume grid procedural interface**
  - generation: `otGrid *otGridGenerate(swrGeometry *)`
  - transformation: `utGrid *utGridfromOt(otgrid *)`
  - queries: `otForEachGridPoint(otGrid *, otGridPoint *)`
  - field values: `otGridAddSolution(otGrid *, char* solName, int volumeIndex...)`
  - adaptation: `otGridAdapt(otGrid *, int adaptFlag, char* elementSolutionName)`

Complete specification at: `http://www-ee.stanford.edu/tcad.html`

Program Flow (diffusion) —— An Illustrative Glimpse

```
1. // construct geometry from VIP-3D or solid modelers

2. VIPtoSWR(); // use PI such as swrGeometryAddVolume()

3. otGrid = otGridGenerate(*swrGeometry); // oct-tree grids

4. otGridAddSolution(*otGrid, "Boron"); // one-species only

5. SWRtoAlamode(*otGrid, *alamodeGrid); // otForEachGridPoint()

6. For (every 20 time steps of AlamodeDoDiffusion(*alamodeGrid))

7.     { AlamodetoSWR(*alamodeGrid, *otGrid);

8.       otGridAdapt(*otGrid, "elementError"); // adaptation

9.       SWRtoAlamode(*otGrid, *AlamodeGrid);  }

10.SWRtoVisualizer(*otGrid); // look at results
```

Current Status of Implementation

- FOREST (2D quad-tree grids)
  - compliant to first trial version of SWR 0.3; full documentation
  - geometry and grid movement
  - interface with solid modeler, SUPREM and ALAMODE
- CAMINO (3D oct-tree grids)
  - compliant to SWR 0.3
  - interface with ALAMODE and PROPHET
- 2D and 3D level-set geometry movement
  - compliant to SWR 0.3
  - interface with SPEEDIE
- EUCLID (3D unstructured tetrahedra)
  - 3D grid refinement and movement

Conclusions

- Strategies for TCAD tool integration involve not only functionality and efficiency, but also practicality
- SWR 0.3 provides a communication and development platform with open interface for both old and new TCAD tools
- Computational prototyping software of hybrid physical systems (such as MEMS) will need further extended functionalities
  - more element types (bricks, prism, etc.) and gridders
  - grid transformation for finite deformation
  - fields on Eulerian grids for microstructure
  - ???