IBM's Integrated Data Model

Joseph K. Morrell
IBM EDA Laboratory
East Fishkill, NY

David J. Hathaway
IBM EDA Laboratory
Burlington, VT

IEEE/DATC Electronic Design Processes Workshop
April 26-28, 2000
Monterey, CA

History

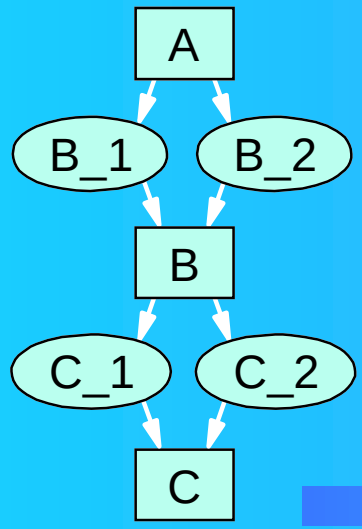
- 70's - Mid 80's
  - File based point tools running on VM/MVS mainframes
  - Limited common model within logic synthesis tools
- Mid 80's - Mid 90's
  - Migration to workstations
  - Synthesis and Timing integrated on common runtime model
  - Floorplanning and Timing integrated on (different) common runtime model
  - Remaining PD still point tools utilizing common files

History *(cont.)*

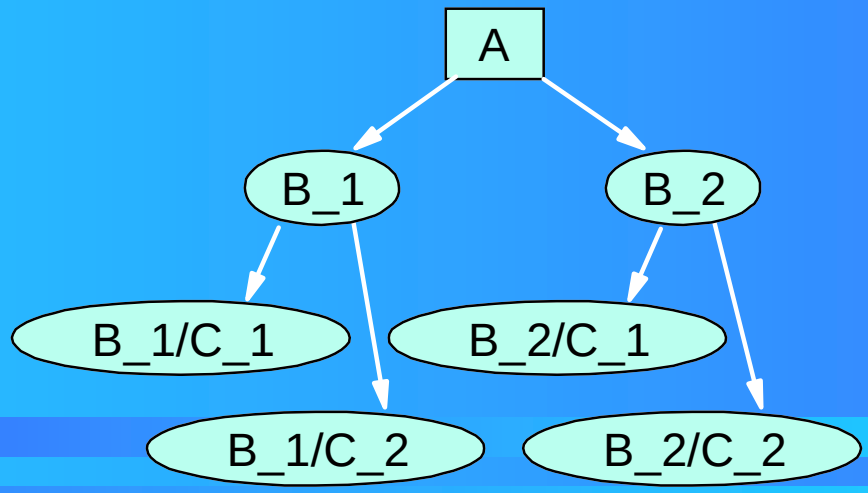
- Mid 90's to now
  - Synthesis, Floorplanning, and Timing migrated to common runtime model
  - Other design tools (e.g. Placement, Wiring) being added to common runtime
  - Other analysis tools (e.g. Extraction, Noise) also being added
  - Drivers
    - Need for tool integration
    - Desire to share common functions
      - More efficient resource utilization
      - Consistency

IDM Structure

- Folded Model
  - Hierarchical Netlist / Structure
  - Cell Definitions and Implementations described once, even when used multiple times
- Occurrence Model
  - Fully expanded structure for instance specific data



Folded model

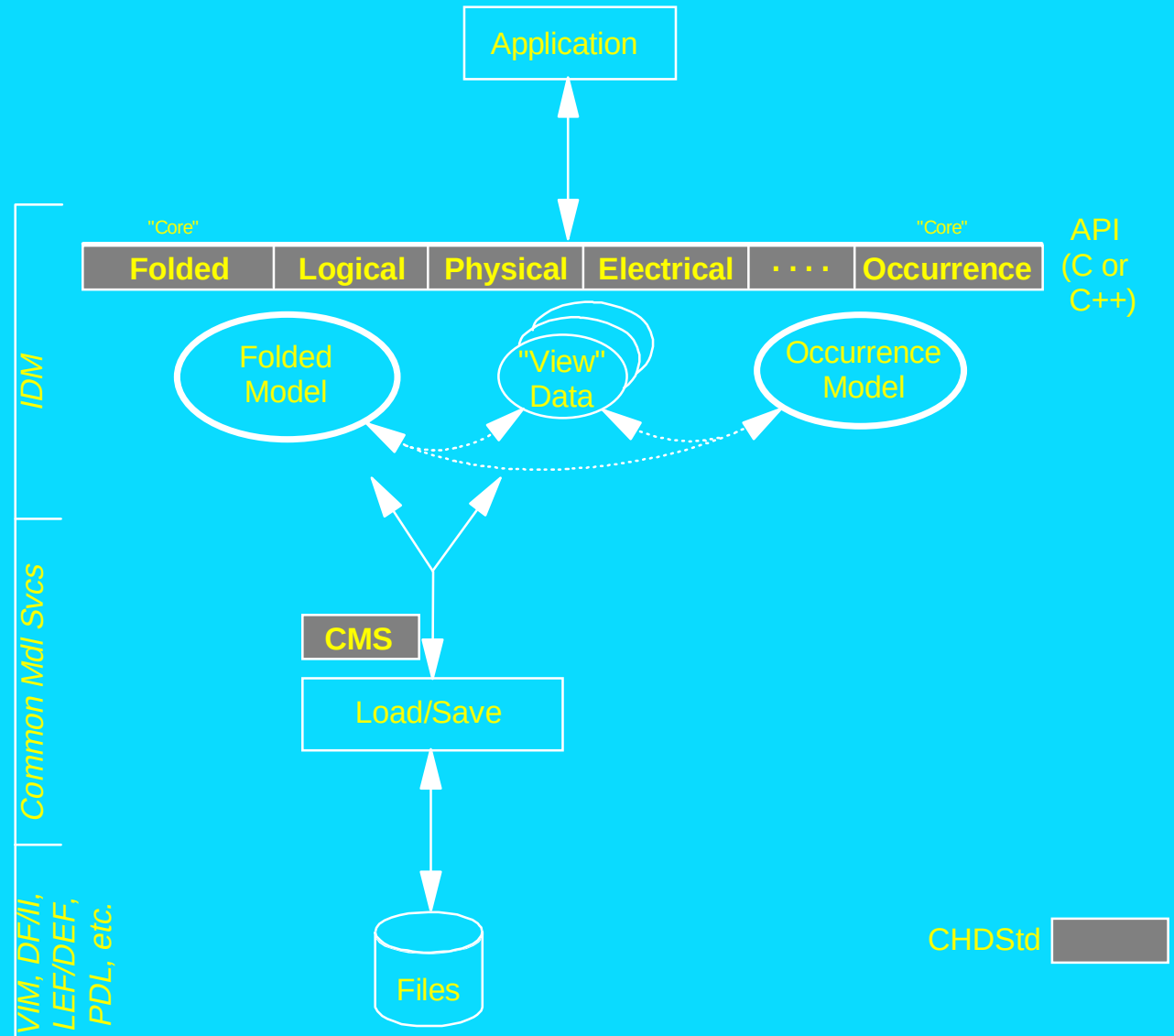


Occurrence model

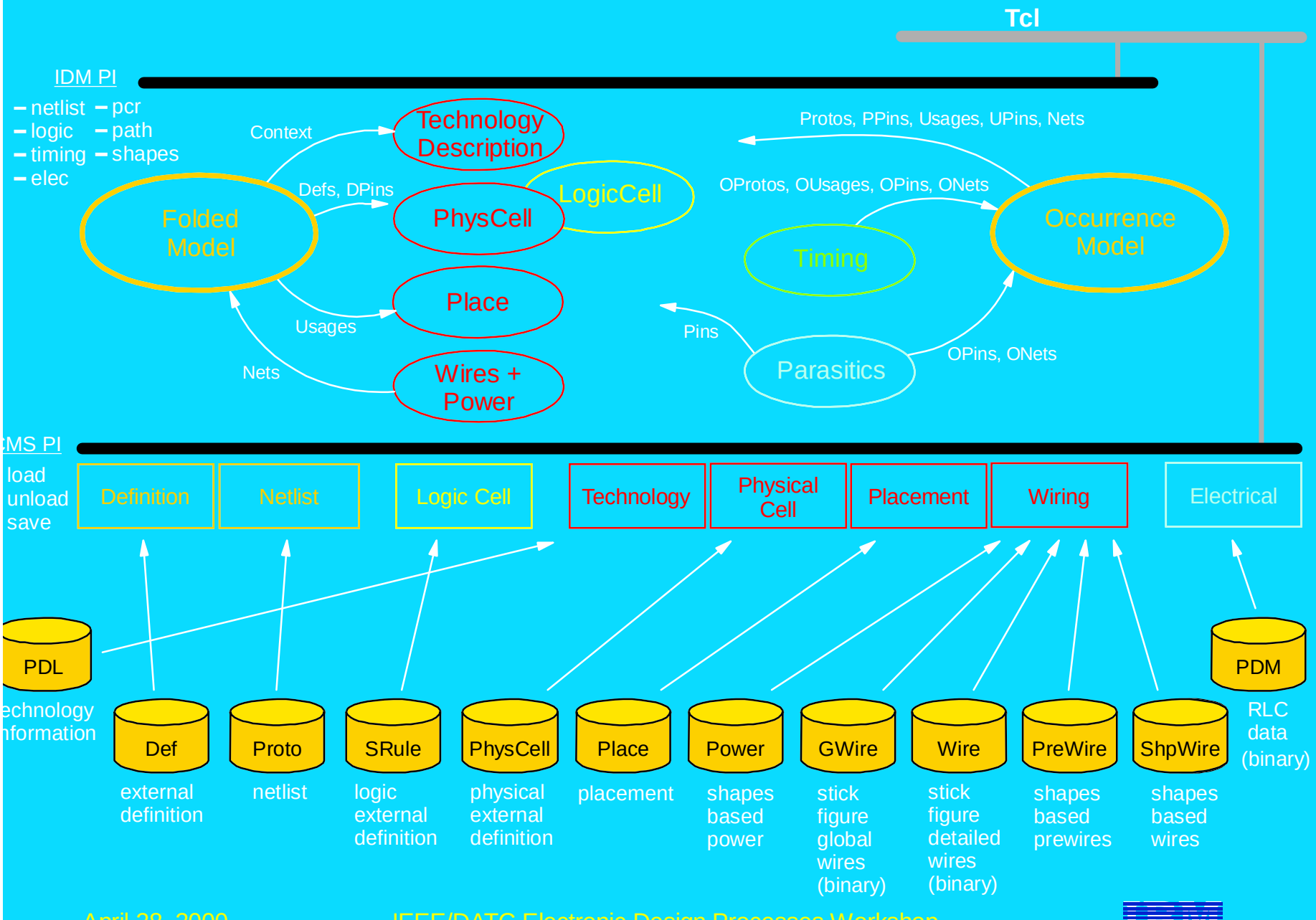
IDM Structure

- View Specific Extensions
  - Rule Box enables applications to extend the model to support view specific external cell descriptions
  - Simple/Complex Property support enables applications to extend the model for other types of view specific data associated with the Folded Model objects, the Occurrence Model objects, or the basic View objects themselves
- Callbacks
  - Granular support to enable application incremental processing
  - Only used for invalidation - lazy evaluation used for recalculation
- Legacy applications supported through thin translation layer to old PIs

IDM Architecture



IDM 0301 Architecture



VIMe: Extended VIM (Persistent Files)

- Hierarchical Netlist - Def / Proto
  - Occurrence Model generated dynamically
- Logic Cell Data - SRule/OLA
- Physical Data - multiple files by data type
  - Physical Cell Rule - PhysCell
    - Shapes-based
    - Outlines, Ports, Blockages, etc.
  - Implementation Data
    - Placement
    - Wiring - stick figure and shapes
    - Explicit voltage net wiring
- Other Files
  - Technology Description - PDL (Physical Design Language)
  - Parasitics - PDM (Parasitic Data Model)
  - Cell Electrical / Delay - IEEE 1481

CMS: Common Model Services

- General set of interfaces for load/save/unload
- Configurable
- Not bound to a particular file type
- Addresses fine granularity of data

Logic View

- PI maintained as part of the IBM Synthesis application
- Synthesis Cell Rule - SRule/OLA
- Logic Edit Capabilities
  - Boolean
  - Registers, Selectors, Adders, Drivers, ...
  - Gates
  - Pins/Ports
  - Networks
  - Equations
  - Size, Power, ...

Physical View

- Same schema as Extended VIM
  - Shape-based (gridded or gridless)
- PCR: Physical Cell Rule
  - Multiple outlines (placement, wiring, etc.)
  - Opens and blockages (placement & wiring)
  - Cktrows and site arrays
  - Ports (collection of shapes for a pin)
  - Tunnels (predefined wiring available for implementing connections of a net)
- Implementation Data
  - Placement
  - Wiring
    - stick figure and shapes
    - signal and power

Technology Description

- PCR Technology Description (base info)
  - Layers & levels
  - Minimum ground rule constraints
  - Electrical characteristics
  - Grids
  - Predefined wire models with wire segments & via models
- PCR Technology Usage (base + design specific info)
  - Layer usages (selection from maximal set)
  - Power specification
  - Design specific constraints
  - Any additional data required, similar to the types allowed in the Tech Desc, except for Layers

Electrical Data

- Electrical Cell Rule - IEEE 1481
- Net Parasitics (ESS - Electrical SubSystem)
  - RLC graph (with coupling)
  - Net / Pin / Load capacitance
  - Resistance
  - Coupling
  - Poles/residues
  - Moments
  - Pi models
  - Delays
- Applications can register extractors/calculators

Timing Data

- PI maintained as part of the IBM Timing application
- Cell Delay Rule - IEEE 1481
- Timing Information
  - Assertions
  - Arrival times (ATs), Required arrival times (RATs)
  - Slacks
  - Slews
  - Delays, tests (setup, hold, pulse width)
  - Phase information
  - Clock tags
  - Loop cuts
  - ...

Tcl Bindings

- Bindings for Folded/Occurrence, CMS, Logic, PCR, Tech Desc/Usage, Timing, ESS
- Write tcl scripts to modify data rather than shell scripts to edit files
- Applications need to address concurrent modifications through, for example, refresh on demand

Example of Application Usage

- Synthesis-Placement-Timing
  - Callbacks are established to monitor interaction
    - Synthesis registers callbacks on placement operations
    - Placement and Timing register callbacks on netlist changes
    - Timing registers callbacks on placement changes
    - Parasitic Estimators / Delay Calculators registered with Electrical View
  - As Placement runs, Synthesis can selectively respond to placement operations by dynamically invoking timing correction operations
  - Timing and Electrical View note objects which have been changed
  - If timing is requested for an object which has been changed, the electrical and/or timing data will be incrementally updated at that time, and returned to the application. Otherwise, the cached data will be returned.

Example of Application Usage