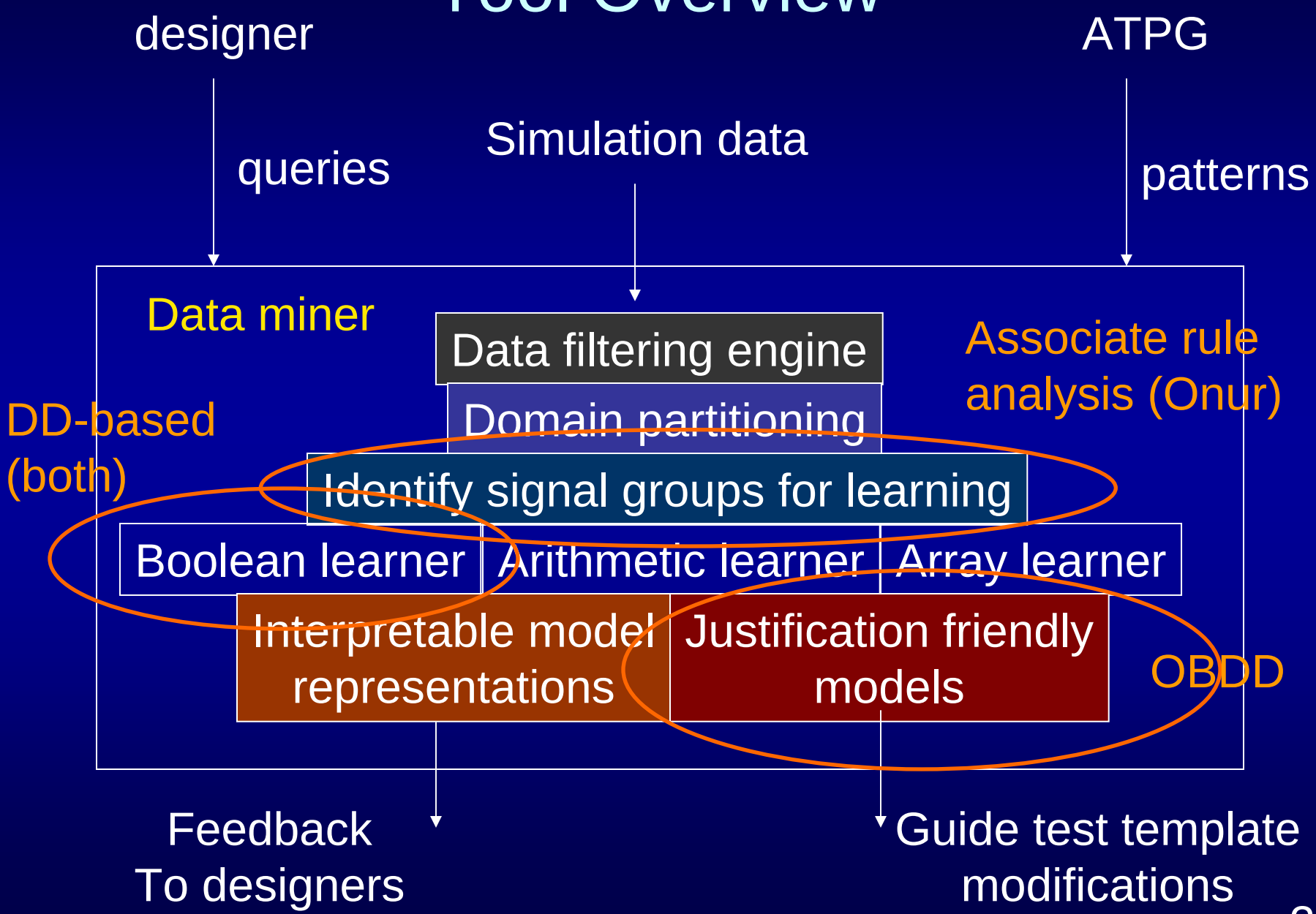
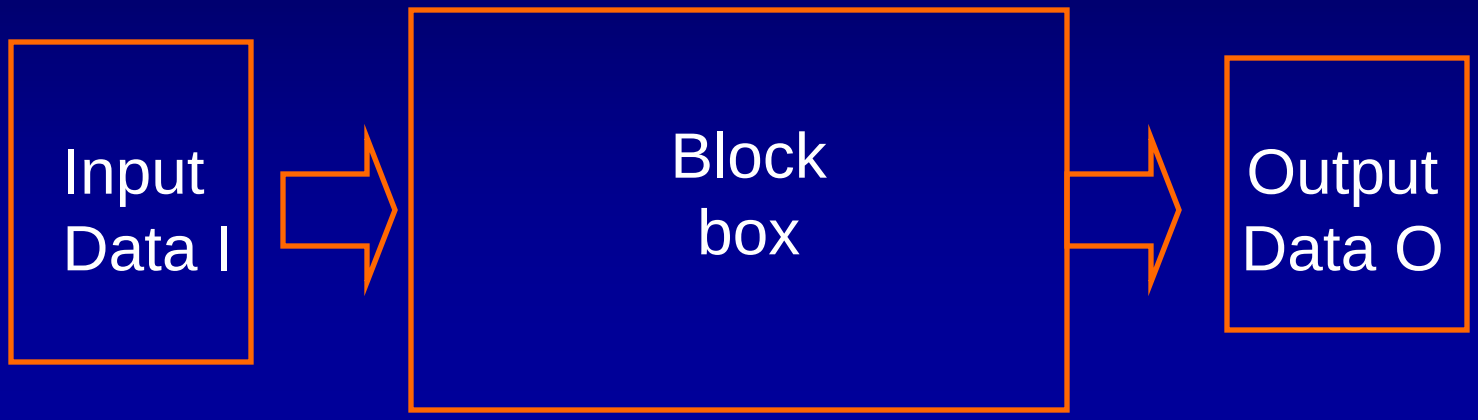
Tool Overview



Boolean learner

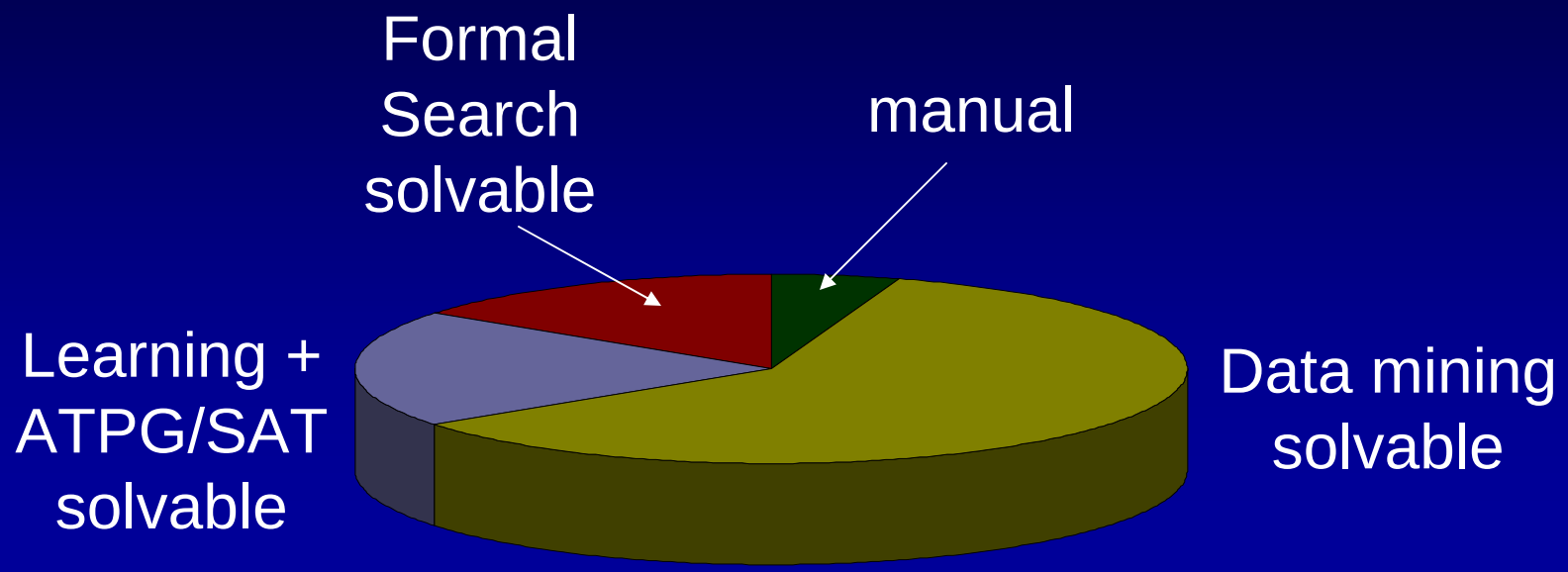
- The basic engine is done and ready to be applied
  - Decision diagram based
  - OBDD representation of learned results
    - Approximate OBDDs
    - Limited size, never blow up
    - Optimized ordering by association rule analysis
  - Comparison to symbolic simulation (Tao Feng)
- Extension to hybrid domain is under way (Onur)

Learner



- Learner will build a model g to relate the I/O behavior: g(I) = O
- Also, given an vector “out”, learner can computes g^{-1}(out) easily

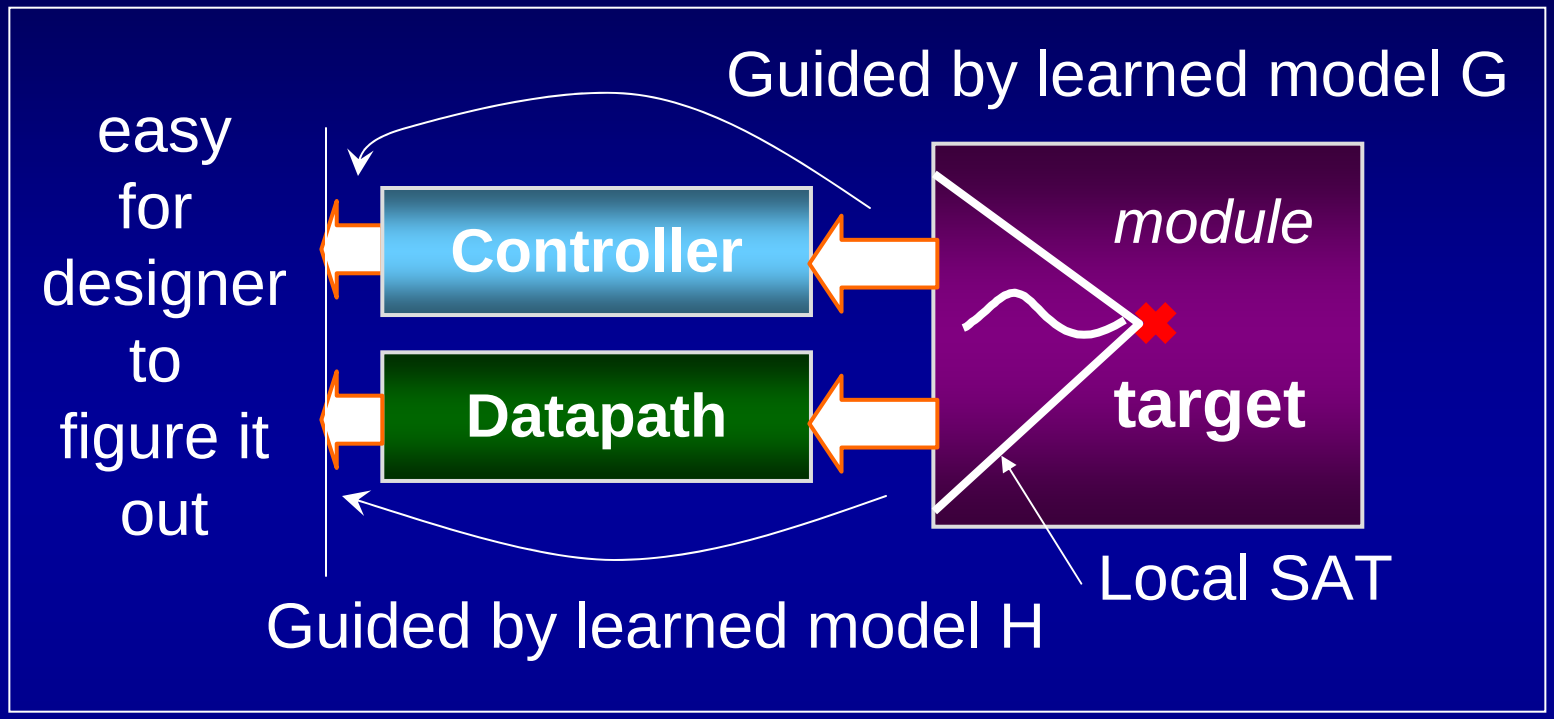
Our Focus



- Our conjecture is that there are many problems that can be solved easily from data learning and mining perspective (but hard for a formal search method)
  - We hope a large portion of the problems are in this category
  - They may be tedious but “not hard”
  - We need to carefully define what problems are in this category

- Learning + formal search may take care of some problems

One Way To Use Learned Models



- Combine learned models with local SAT to guide pattern justification
- Learned models can be on different domains
  - Boolean, Arithmetic, Array

This Vs. Other Methods

- Learning doesn't solve the justifiability problem
  - Given an output pattern, does it exist an input pattern to justify this output pattern?
  - Learning will give you an answer, but without guarantee
  - Completely solving the problem requires search (SAT)
- Design size is irrelevant
  - Design is a block box to learning
  - No internal OBDD blow up problem likes that in symbolic sim.
- Learned OBDDs are approximate OBDDs
  - Sizes are pre-limited
  - Orderings are pre-determined
  - They may be meaningless
    - But in general, we have seen very good results

Designs

- Synthesis benchmarks
- OpenRISC 1200
  - A small processor core
  - 5-stage pipeline
  - Show coverage improvements with proposed learning techniques
- Freescale test-cases to be discussed
- Integrate with industrial test bench development environment
  - Synposys Vera

Discussion – Commonly asked questions

- How can data mining or learning help me to cover an assertion based on a system state that requires 1000 cycles to reach?
  - We don't know yet …
  - There may be problems more suitable for using formal search like SAT or hybrid solver, etc.
  - Our goal is to understand and draw that boundary between problems more suitable for using learning and problems suitable for using formal search
- How can you guarantee your data mining or learning results?
  - Data mining by nature has no guarantee
  - Hence, the focus is on doing as much as possible, as quickly as possible, and know when the tool could fail
- Can your tool help me to generate functional tests for speed binning?
  - Definitely. This may be the one of the applications for industrial experiments
- What if your tool fails?
  - Then, you need to go back to formal search or manual approach
  - But, if the tool fails, you will know it quickly so that you won't waste your time waiting for the results