Configuring Software Product Lines using Clafer and multi-objective optimization

Rafael Olaechea

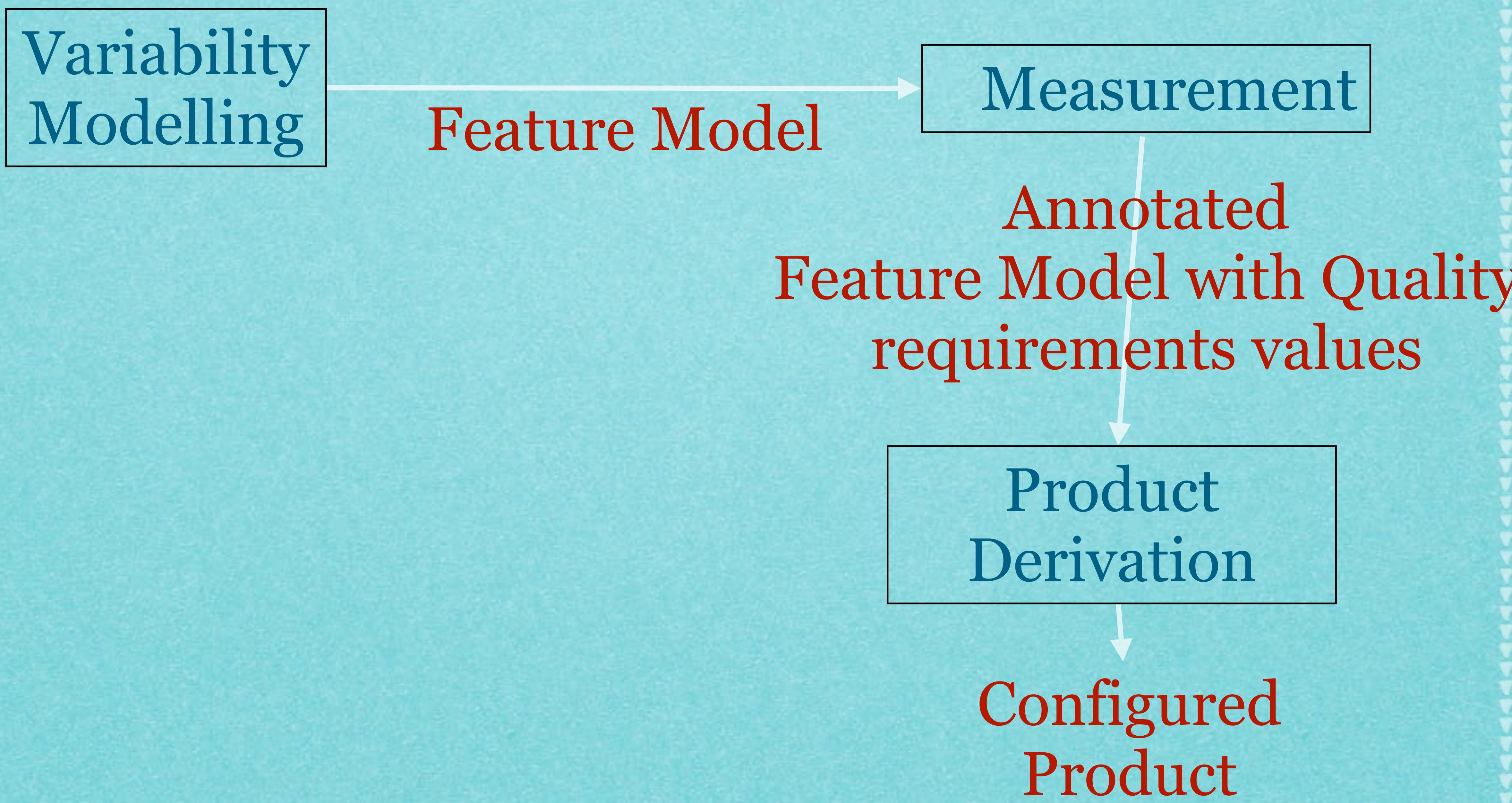
Outline

- Background.
- Methodology including Research Problem.
- Tool
- Evaluation.
- Future Work.
- Conclusions.

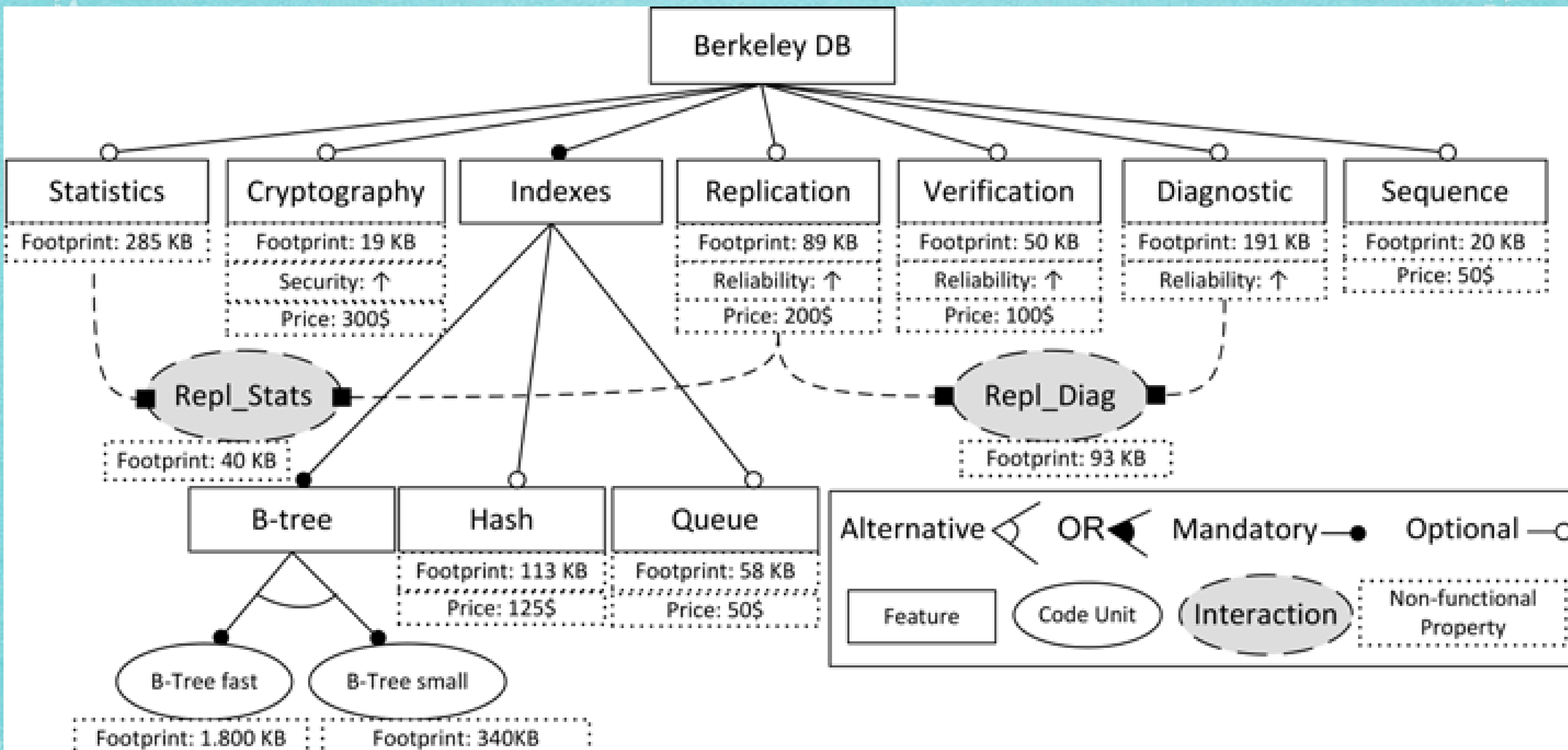
Software Product Lines

- Create family of software systems to be used in a specific domain.
  - Domain Model.
  - Reusable Assets.
  - Configuration Model:
    - Feature Model.
    - Product Derivation Process
- Examples: Medical Imaging Systems Software.

Product Derivation Process



Berkeley DB Annotaed Feature Model with Quality Requirements



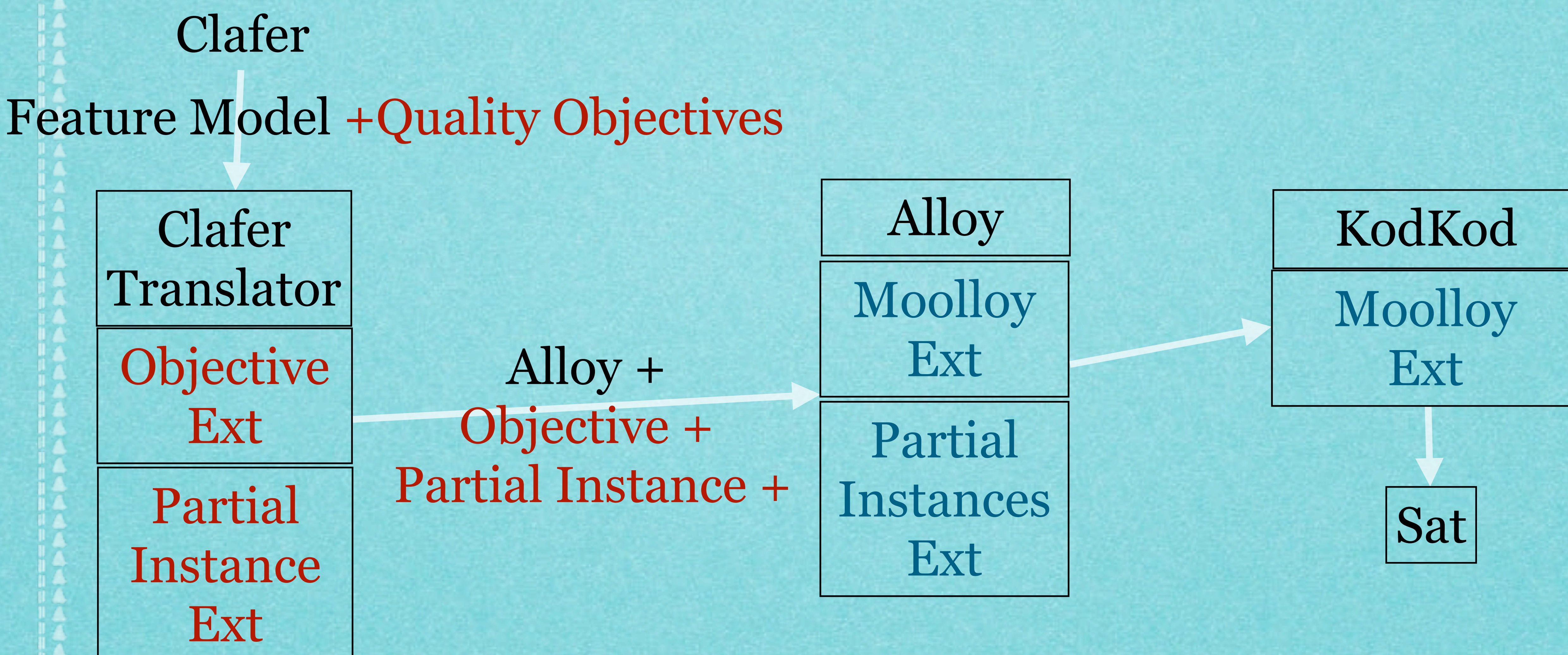
Methodology

- Goal: Improve the product derivation process in Software Product Lines.
- 1. Collect SPL feature model examples.
- 2. Build on existing solutions (Clafer, Moolloy, Alloy partial instances)
- 3. Create extension to Clafer translator.
- 4. Evaluate Performance of tooling.

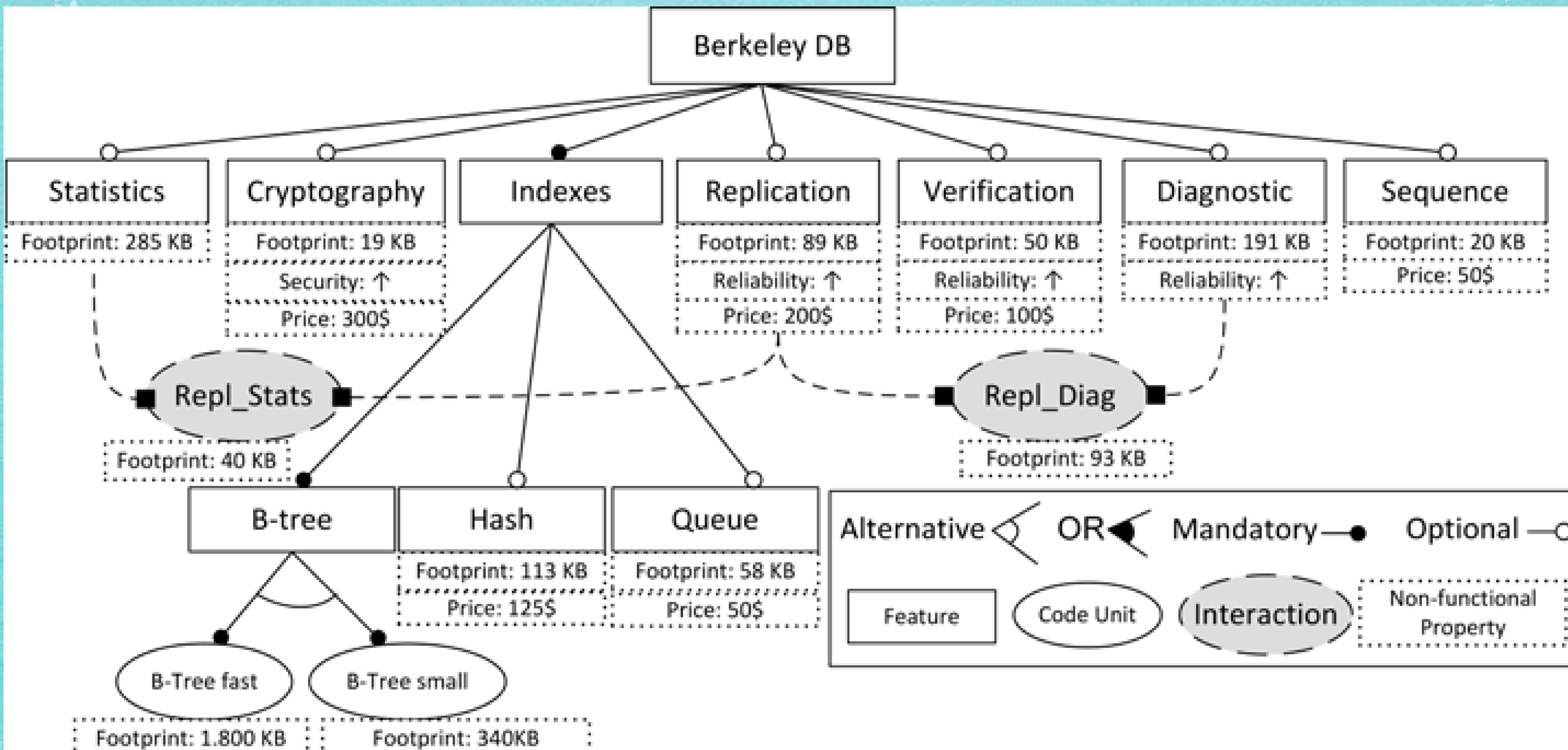
SPL Annotated Feature Models

- Annotate feature models with Quality Requirements:
  - Binary Footprint.
  - Performance
  - Code Complexity
  - Reliability

Tooling Pipeline



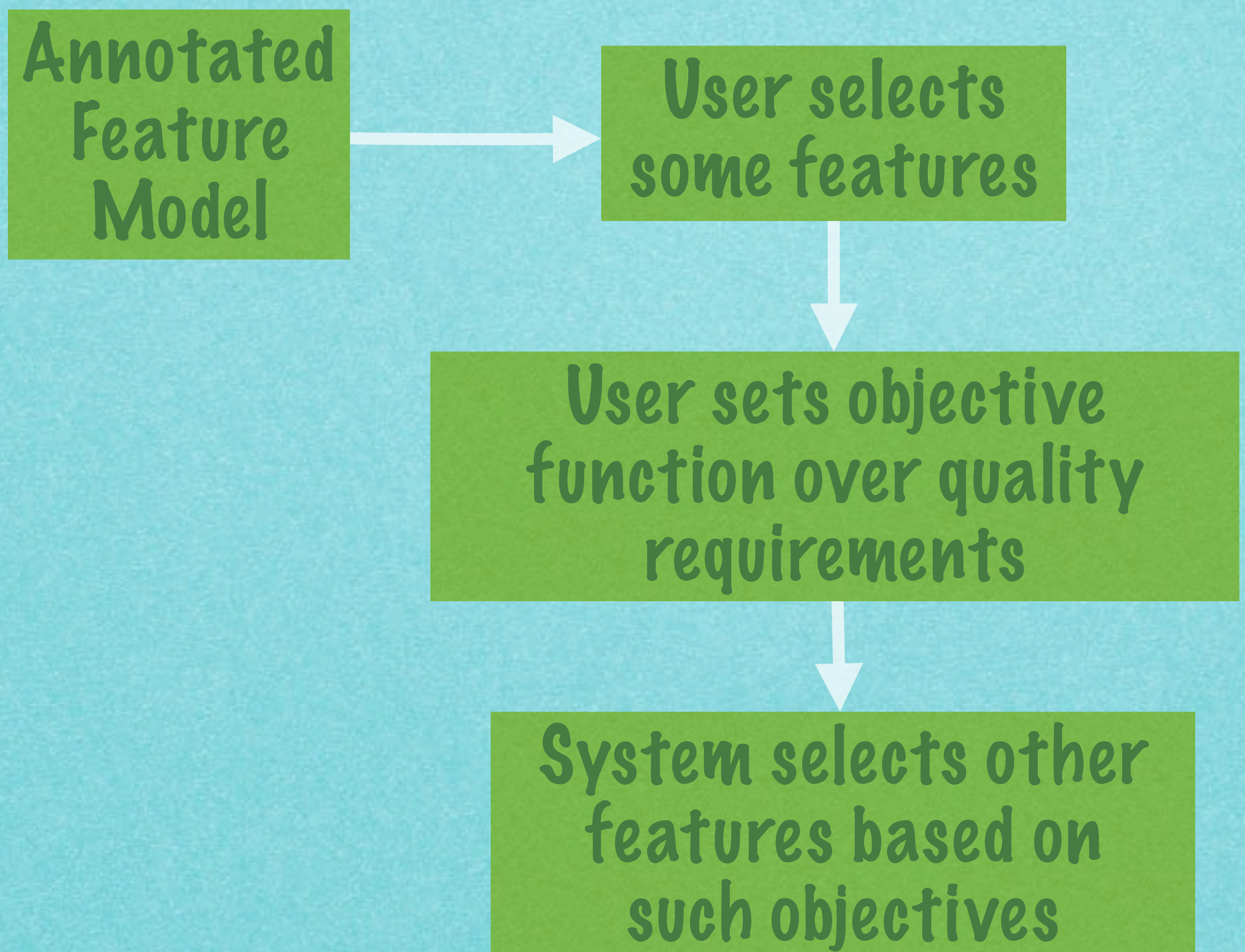
Berkeley DB Annotaed Feature Model



Berkeley DB Feature Model in Clafer

```
abstract BerkeleyDbC
    STATISTICS : IMeasurable  ?
        [ this.footprint = 285]
    CRYPTO : IMeasurable  ?
        [ this.footprint = 19]
    INDEXES : IMeasurable
        [this.footprint = 0]
        xor BTREE
            [this.footprint = 0]
            BTREE_SMALL : Imeasurable
                [ this.footprint = 340]
            BTREE_FAST : IMeasurable
                [ this.footprint = 1800]
        HASH : IMeasurable  ?
            [ this.footprint = 113]
        QUEUE : IMeasurable  ?
                [ this.footprint = 58]
    REPLICATION : IMeasurable  ?
        [ this.footprint = 89]
    (...)
```

Optimizing Quality Requirements Workflow



Evaluation on Sample Feature Models

SPL	Features	Time (s)	Binary Footprint (kB)
LinkedList	18	71	4.43
LinkedList [Print and Measurement]	18	21	10.64
SQLite	80	32278	1200
Berkeleydb	8	23.6	1804

Other Feature Models

- Violet UML - UML Diagramming Tool, ~ 200 features.
- ZipMe - Zipping program.
- Prevayler - Java Persistence framework.
- PKJab - Instant Messenger Application.
- Apache - Web Server.
- BerkeleyDB Java Version - Database.

Extending Clafer with Objectives: LinkedList Feature

```
abstract LinkedList
    xor AbstractElement : IMeasurable
        [ this.footprint = -12]
        ElementA : IMeasurable
            [ this.footprint = 12]
        ElementB : IMeasurable
            [ this.footprint = 0]
        (...)
    xor AbstractSort : IMeasurable  ?
        [ this.footprint = 57]
        BubbleSort : IMeasurable
            [ this.footprint = 17]
        MergeSort : IMeasurable
            [ this.footprint = 32]
        (...)
    print : IMeasurable  ?
        [ this.footprint = 44]
    Measurement : IMeasurable  ?
        [ AbstractSort ]
        [ this.footprint = 484]
        (...)
    total_footprint :integer =
  sum(IMeasurable .footprint)
```

```
abstract IMeasurable
    footprint : integer

config : LinkedList
   [print && Measurement]
<< min config.total_footprint >>
<< max config.total_performance>>
```

Extending Clafer with Objectives

- LL_Configuration: LinkedList_FeatureModel
  - [print && Measurement]
- **<< min LL_Configuration.total_footprint >>**
- **<< max LL_Configuration.performance >>**

Optimizing Footprint + Performance

- objectives o_global { **minimize** [c229_simpleConfig.@r_c121_total_footprint.@c121_total_footprint_ref] , **maximize** [c229_simpleConfig.@r_c122_total_performance.@c122_total_performance_ref] }
- Get a set of solutions in the optimal front between performance and footprint.

Reasoning Optimization: Partial Instances in Alloy

- Alloy Extension to express scope in terms of concrete instances.

-Clafer translator generates a partial instance block to improve performance of reasoning in alloy.

```
inst partialinstance {
  12 int,  // bitwidth
  relation_footprint in ...
}
```

Optimizing Footprint

- **Translate Clafer Objectives into Alloy:**
  - objectives o_global { minimize [c229_simpleConfig.@r_c121_total_footprint.@c121_total_footprint_ref] }

Execute using Multi-Objective Alloy:

Found base solution. At time: 3, Improving on [586]
Found a better one. At time --: 3, Improving on [586]
Found a better one. At time --: 6, Improving on [467]
Found a better one. At time --: 27, Improving on [444]
Found a better one. At time --: 43, Improving on [443]
GIA ----: [443]

Future Work

- Integrate Sparse Integer Support from Alloy.
  - Breadth-Width Search could create set of all reachable integers.
- Integrate partial non-optimal results from the alloy solver before reaching the optimal answers.
  - For Sqlite it took 13 hours, but last 7 hours gave only marginal improvement.

Challenges

- Partial Instances in Alloy:
  - Ongoing work from Vajih Montaghami.
- Getting Realistic Software Product Line Models
  - Wrote translator to get real models from SPLConqueror work by Norbert Siegmund et al.

Conclusions

- Product Configuration in Clafer
  - Explore Space of Product Configurations
  - Helps Stakeholders consider quality properties.
- Quality of Annotated Feature Models.

References

- SPL Conqueror: Toward optimization of Non-functional Properties in Software Product Lines. N. Siegmund et Al. Software Quality Journal.
- Extending Alloy with Partial Instances. V. Montaghami, D. Rayside.
- Scalable Prediction of Non-functional Properties in Software Product Lines. N. Siegmund et Al. SPLC 2011.
- The Guided Improvement Algorithm for Exact, General-Purpose, Many-Objective Combinatorial Optimization. Rayside et al.
- Feature and Meta-Models in Clafer: Mixed, Specialized and Coupled. K. Bak, K. Czarnecki, Andrzej Wasowski.