

Model Clone Detection

Finding duplicate or similar fragments in higher-level representations

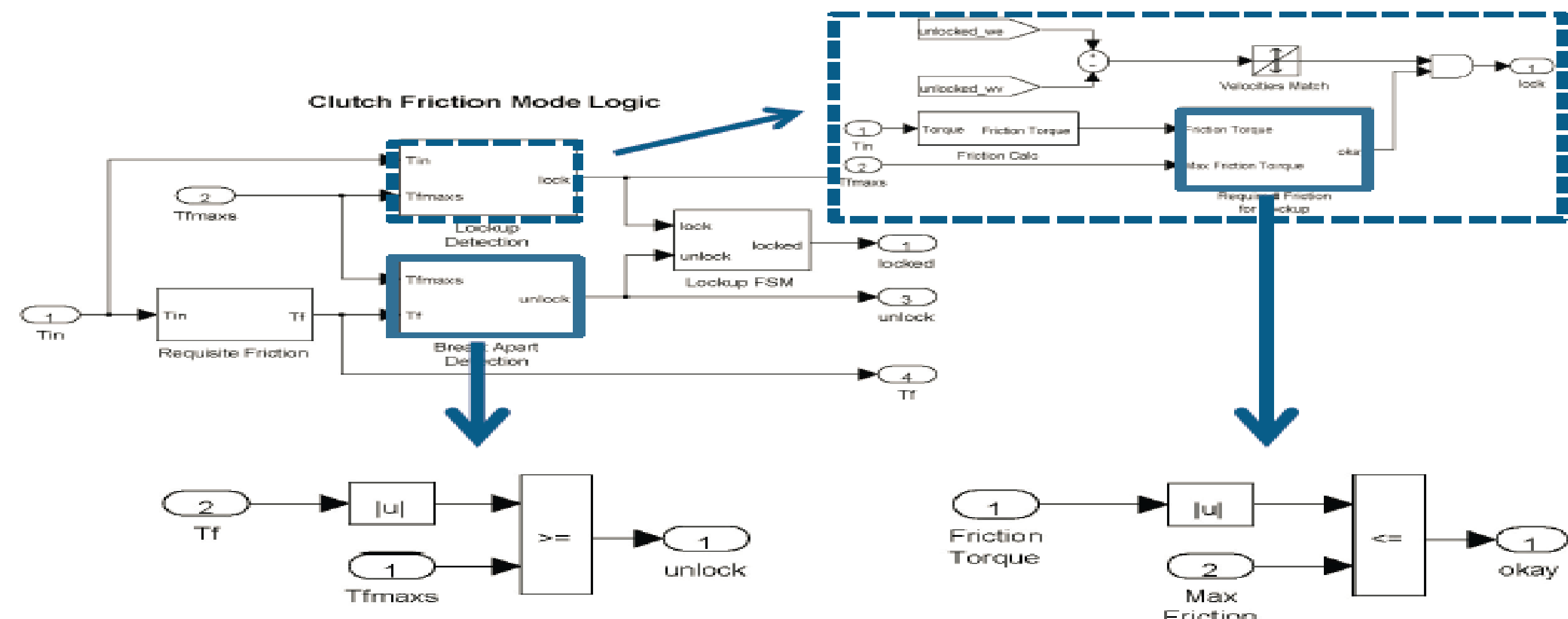


Figure 1. Example from Simulink automotive example model set showing, like code clones, model clones can cross structural and hierarchical levels.

Initial Results

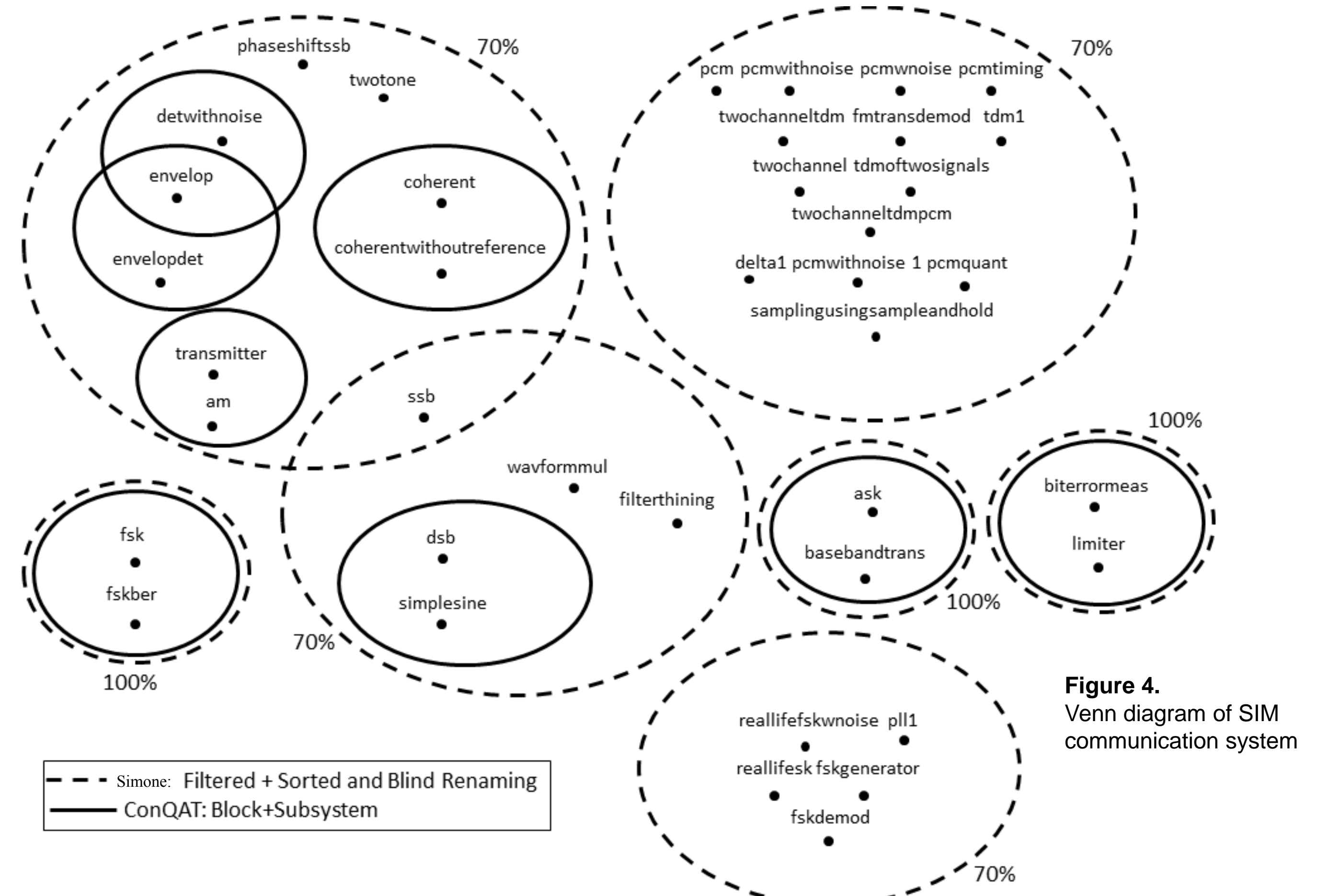


Figure 4. Venn diagram of SIM communication system

Existing Approaches

- Graph-Based
 - ConQAT
 - eScan,aScan
 - Naïve Clone Detector

- Text-Based
 - Simone (Our Tool)

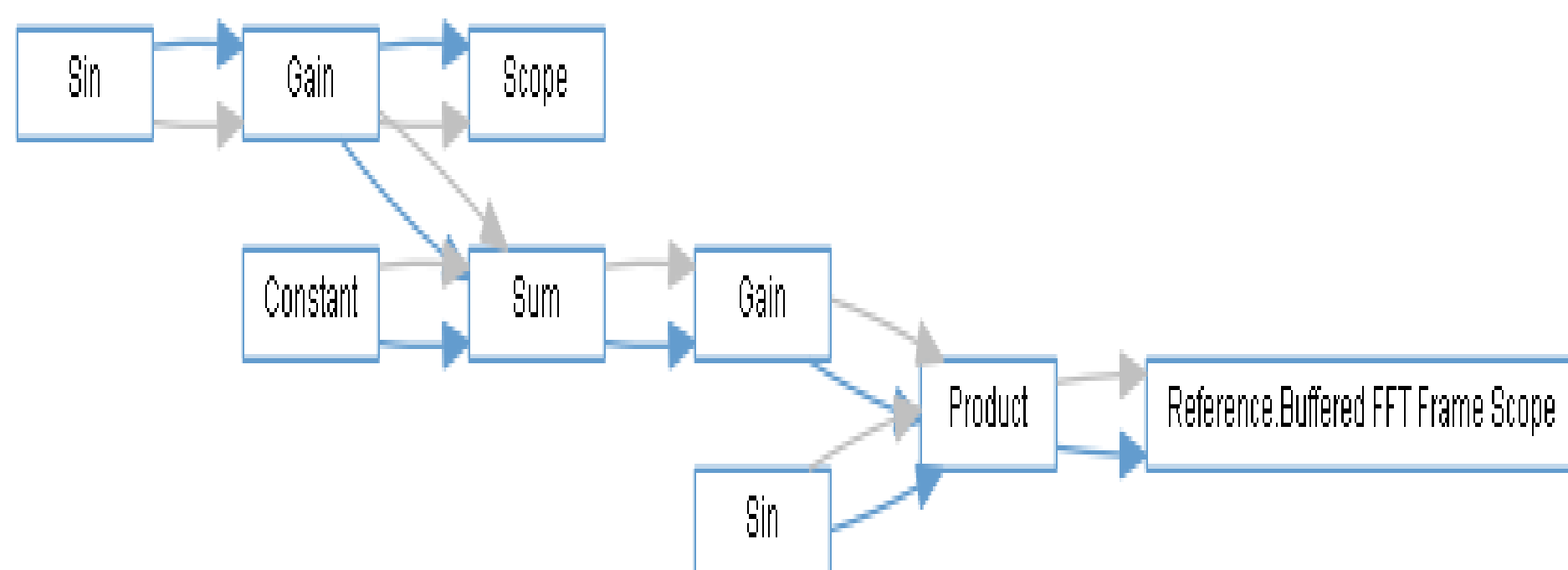


Figure 2. Graphical representation of model clone

```

System {
  Name          *onoff*
  Location      [168, 385, 668, 686]
  Open          on
  ModelBrowserVisibility off
  ModelBrowserWidth 200
  ScreenColor   *automatic*
  PaperOrientation *landscape*
  PaperPositionMode *auto*
  PaperType     *usletter*
  PaperUnits    *inches*
  ZoomFactor    *100*
  AutoZoom     on
  ReportName    *simulink-default.rpt*
  Block {
    BlockType   DiscretePulseGenerator
    Name        *DiscretePulseGenerator*
    Position    [45, 25, 75, 55]
    Amplitude   *1*
    Period      *2*
    PulseWidth  *1*
    PhaseDelay  *0*
    SampleTime  *1*
  }
  Block {
    BlockType   Product
    Name        *Product*
    Ports       [2, 1, 0, 0]
    Position    [145, 67, 175, 98]
    Inputs      *2*
    SaturateOnIntegerOverflow on
  }
}
    
```

Figure 3. Textual representation of a Simone model system clone

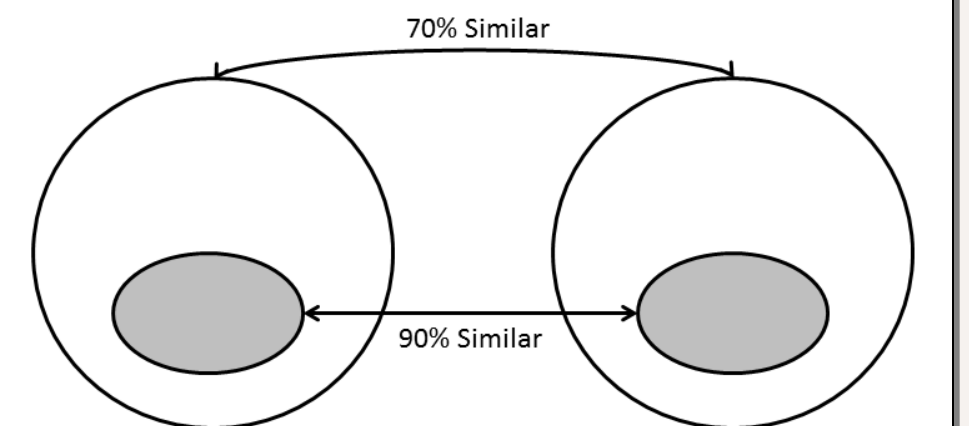
Challenges Faced

Recall

Of all the system clones in existence, how many of them will be found by each respective tool.

Nature

Nature of reported clones may differ:
 *One enforces (sub-)system boundaries while the other does not;
 *Nested clones as shown in Figure ->



Representation

The representation of the resulting clone classes and instances from each tool may be different and require some form of normalization

Purpose of Investigation & Evaluation Areas

Strengths

Relevance

Weaknesses

Performance

Evaluate Simone

Particularly interested in which methods are better suited to identify frequent sub-model patterns in large model sets, a goal of our industrial partners.

Clone Detection Type

User Interaction Required

Adaptability

Model Pattern Granularity

Proposed Solution – Extend Framework with Mutation Analysis

Use mutation operators that will introduce variations of the 3 clone types and look for the resulting mutations, explicitly. Some sample mutation operators we have begun working with include:

Rename Block or Line

Changing a Block's Value

Changing a Block's Type

Adding or Deleting a Source Block

Adding or Deleting a Destination Block

Model Sets

Publicly Available Models {

- "Matlab Central" Repository
- 5 Projects + Others

Industrial Partner's Models {

- Automotive Industry

Acknowledgements

This work is supported in part by NSERC, as part of the NECSIS Automotive Partnership with General Motors, IBM Canada and Malina Software Corp. Benjamin Hummel has been very helpful in providing advice and the assistance with model sets and ConQAT.