A Template for Predictability Definitions with Supporting Evidence

Daniel Grund\textsuperscript{1}  Jan Reineke\textsuperscript{2}  Reinhard Wilhelm\textsuperscript{1}

\textsuperscript{1}Saarland University, Saarbrücken, Germany
\textsuperscript{2}University of California, Berkeley, USA

Workshop on Predictability and Performance in Embedded Systems
Outline

1. Motivation & Problem
2. Key Aspects of Predictability
3. Supporting Evidence
4. Summary
Outline

1 Motivation & Problem

2 Key Aspects of Predictability

3 Supporting Evidence

4 Summary
Motivation

The Platitude

Complexity of embedded systems increases. They become more and more “unpredictable.”

- **Timing analysis**
  - current methods will not scale to future systems
  - need better analyses or more predictable systems

- **Projects**
  - **PREDATOR**: design for predictability and efficiency
  - **PRET**: reintroduce timing predictability and repeatability
  - **MERASA**: guarantee analyzability and predictability
The Problem

- We don’t know what predictability actually is or should be
- People mean different things when saying “predictability”
- Criteria for predictability are mostly
  - intuitive
  - case-based
  - phenomenological, symptom-based
Examples

<table>
<thead>
<tr>
<th>Predictability</th>
<th>better</th>
<th>worse</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU pipeline</td>
<td>in order</td>
<td>out of order</td>
</tr>
<tr>
<td>branch prediction</td>
<td>static</td>
<td>dynamic</td>
</tr>
<tr>
<td>cache replacement</td>
<td>LRU</td>
<td>FIFO</td>
</tr>
<tr>
<td>scheduling</td>
<td>static</td>
<td>dynamic preemptive</td>
</tr>
<tr>
<td>arbitration</td>
<td>TDMA</td>
<td>FCFS</td>
</tr>
</tbody>
</table>

- This is mostly based on intuition
- Is there a common underlying principle?
Vision

- Derive formal definition of predictability
- In the ideal case
  - uniformly applicable
  - rather constructive than existential
Outline

1 Motivation & Problem

2 Key Aspects of Predictability

3 Supporting Evidence

4 Summary
What is Predictability?

Oxford Dictionary

<table>
<thead>
<tr>
<th>predictable</th>
<th>adjective, able to be predicted</th>
</tr>
</thead>
<tbody>
<tr>
<td>to predict</td>
<td>verb, state that a specified event will happen in the future</td>
</tr>
</tbody>
</table>

- Want to consider hardware/software systems
- That is, deterministic and finite systems
- Concretize above definition for such systems
Aspect 1: Property to be Predicted

- System behaviors can be described by a set of traces
- Exact behavior often irrelevant
- Derived properties are interesting

⇒ Which property of the system to predict?
  - number of certain events on a trace
  - maximal length of traces (WCET)
Aspect 2: Sources of Uncertainty

- Deterministic finite systems
  - Any property can be determined exactly
- However, properties depend on something unknown
  - Predictions shall hold in any case

⇒ What are the sources of uncertainty?
  - program input → execution time
Aspect 3: Quality Measure

- Generally, predictability is not a Boolean property
- Allow shades of gray
- How well can a property be predicted?

⇒ What is the quality measure on predictions?
Aspect 4: Inherence

Consider

- two systems \( S \) and \( T \)
- an analysis \( A \) for a system property \( P \)

Assume \( A \) can determine \( P \) for \( S \) better than for \( T \)

Should \( S \) therefore be more predictable than \( T \)?

We say no; there could be \( A' \) where it is vice versa

\( \Rightarrow \) Predictability should be a property inherent to the system
Proposition

The notion of predictability should capture if, and to what level of precision, a specified property of a system can be predicted by an optimal analysis.

- In particular, a definition should state
  - Property to determine
  - Sources of uncertainty
  - Quality measure
Example: Timing Predictability

Definition (Timing Predictability)

\[
\Pr_p(Q, I) := \min_{q_1, q_2 \in Q} \min_{i_1, i_2 \in I} \frac{T_p(q_1, i_1)}{T_p(q_2, i_2)}
\]

- Property: execution time of a program
- Sources of uncertainty: program input & hardware state
- Quality measure: variance in execution time
- Measure is within [0..1]
- 1 means perfectly predictable
Outline

1 Motivation & Problem

2 Key Aspects of Predictability

3 Supporting Evidence

4 Summary
Supporting Evidence?

- Work on architectural components that are better predictable
- Usually based on sensible, yet informal intuitions
- Try to cast that work in terms of our template
Examples — Part 1

- Lickly et al.: Pred. programming on a precision timed arch. (PRET)
- Bhat & Mueller: Making DRAM refresh predictable
- Barre et al.: A predictable SMT scheme for hard real-time

<table>
<thead>
<tr>
<th>Hardware unit</th>
<th>Property</th>
<th>Source of uncertainty</th>
<th>Quality measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thread-interleaved pipeline and scratchpad memories</td>
<td>Execution time</td>
<td>Uncertainty about initial state and execution context</td>
<td>Variability in execution times</td>
</tr>
<tr>
<td>DRAM controller</td>
<td>Latency of DRAM accesses</td>
<td>Occurrence of refreshes</td>
<td>Variability in latencies</td>
</tr>
<tr>
<td>SMT processor</td>
<td>Execution time of tasks in real-time thread</td>
<td>Uncertainty about execution context, i.e., other tasks executing in non-real-time threads</td>
<td>Variability in execution times</td>
</tr>
</tbody>
</table>
Examples — Part 2

- Bodin & Puaut: A WCET-oriented static branch prediction scheme
- Schoeberl et al.: Towards time-pred. data caches for chip-MP
- Rochange & Sainrat: A time-pred. exec. mode for superscalar pipelines

<table>
<thead>
<tr>
<th>Hardware unit</th>
<th>Property</th>
<th>Source of uncertainty</th>
<th>Quality measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Branch predictor</td>
<td>Number of branch mispredictions</td>
<td>Analysis imprecision (Uncertainty about initial predictor state)</td>
<td>Statically computed bound (Variability in mispredictions)</td>
</tr>
<tr>
<td>Memory hierarchy</td>
<td>Number of data cache hits</td>
<td>(Percentage of accesses that can be statically classified)</td>
<td>Among others, uncertainty about addresses of data accesses</td>
</tr>
<tr>
<td>Superscalar out-of-order pipeline</td>
<td>Execution time of basic blocks</td>
<td>Analysis imprecision (Uncertainty about the pipeline state at basic block boundaries)</td>
<td>Qualitative: analysis practically feasible (Variability in execution times of basic blocks)</td>
</tr>
</tbody>
</table>
Discussion

- 13 works on predictability improvements tabulated in paper
- Most fit well into the predictability template
- Parameters are indeed instantiated differently
- Many approaches disagree on quality measure
  - evaluation of predictability improvement using analyses
  - practical approach
  - not conclusive on (inherent) predictability
Outline

1 Motivation & Problem

2 Key Aspects of Predictability

3 Supporting Evidence

4 Summary
Summary

- Key aspects of a predictability definition:
  - Property to be predicted
  - Sources of uncertainty
  - Quality measure
  - Inherence

- Predictability template provides mental frame
- Most works on predictability can be cast as template instance