Towards a Common WCET Annotation Language: Essential Ingredients

Raimund Kirner, Albrecht Kadlec and Peter Puschner
Adrian Prantl, Markus Schordan and Jens Knoop

Institute of Computer Engineering,
Institute of Computer Languages,
TU Vienna

8th Int’l Workshop on Worst-Case Execution Time Analysis (WCET’08)
July 1, 2008
Prague - Czech Republic
Why a Common Annotation Language?

**Specifying the problem-inherent information**

- Tool-independent
- Methodology-independent

**Supporting different kinds of analysis**

- Static analysis
- Measurement-based methods

**Expressive enough to support annotation**

- At source code level
- At object code level

Enable inter-operation of tools and increase productivity.
Our Treasure Hunt

Identification of the Essential Ingredients

- Investigated timing-annotation constructs in existing languages and tools.
- Identified the essential ingredients of each timing-annotation and categorized the found ingredients.
- Described the ingredients conceptually to allow instantiation for different representation levels and tools.
Essential Ingredients

Annotation Categorization \([L,C,G]\]
- Layer, class, group.

Addressable Units \([U]\]
- Control-flow addressable units, loop contexts, call contexts, values of input variables, explicit enumeration of infeasible paths, goto statement.

Annotation Information \([I]\]
- Software-Specific Annotation Information
  - Loop bounds, linear flow constraints, variable restrictions, summaries of external functions, unreachable code
  - Predicate evaluation, control-flow reconstruction.
- Hardware-Specific Annotation Information
  - Memory and memory accesses, absolute time bounds.
Every timing constraint is associated with one or more annotation layers.
Timing Invariant

A timing constraint is a *timing invariant* at its associated annotation layer $L$, iff for all possible systems that use annotation layer $L$, it holds that for all possible initial system states the system execution fulfills the timing constraint $C$.

Timing Fiction

A timing constraint is a *timing fiction* if it allows to query alternative program behavior. (→ it must be marked as such)

Conflict

In the case that *timing invariants* and *timing fictions* are in conflict, the semantics of *timing fictions* is to override conflicting *timing invariants*.
Example

```c
void f(int a, char[] b)
{
    int i;
    a = a % 20;
    for (i = 0; i < a; i++)
    {
        //loop1
        if (i % 2 == 0)
            b[i] = a;  //m1
        else
            b[i] = 0;  //m2
    }
}
```

Timing Invariant:
Expressing as linear flow constraint that the then-path is executed at least as often as the else-path: $m_1 \geq m_2$

Timing Fiction:
Specifying a lower and upper loop bound of 40:
$LB(loop1) = 40 \ldots 40$
How to use the ingredients

Annotation = (L, C, G, U, I)

L ... Layer (e.g. program layer, operation layer, ...)
C ... Class (timing invariant or timing fiction)
G ... Group (for practical purposes)
U ... Addressable Unit (possibly referenced)
I ... Annotation Information
Ideal Work Flow

TOOL

Input Code

Front End Annotation Parser

WCET Analyzer

Annotation Generator

Annotated Code

Manual Annotation by User

No

WCET ok?

Yes

Annotated Code

WCET Analysis Result

Markus Schordan (TU Vienna) | Annotation Language Ingredients
Conclusions

A common annotation language including all essential ingredients may allow ...

- Evaluation of tools by using the same set of annotated input sources.
- Language designers and tool developers to focus separately on the essential ingredients of timing annotations, $(L, C, G, U, I)$.
- Create interchangeable building blocks within a timing analysis framework (as intended by ARTIST2).
Thank you for your attention!

CoSTA
Austrian Science Fund
contract P18925-N13
costa.tuwien.ac.at

ALL-TIMES
7th EU R&D FP
contract 215068
all-times.org

ARTIST2 NoE
6th EU R&D FP
contract IST-004527
artist-embedded.org