METAMOC: Worst-Case Execution Time of Embedded Software on ARM Processors

Martin Toft
mt@cs.aau.dk

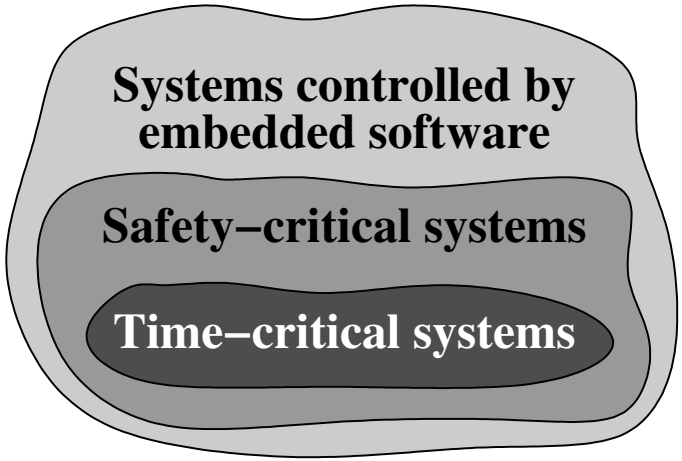
PhD student
Distributed and Embedded Systems
Department of Computer Science
Aalborg University

Joint work with Mads Christian Olesen, Andreas Engelbredt Dalsgaard,
Kim Guldstrand Larsen and René Rydhof Hansen
{mchro,andrease,kgl,rrh}@cs.aau.dk

May 21, 2010

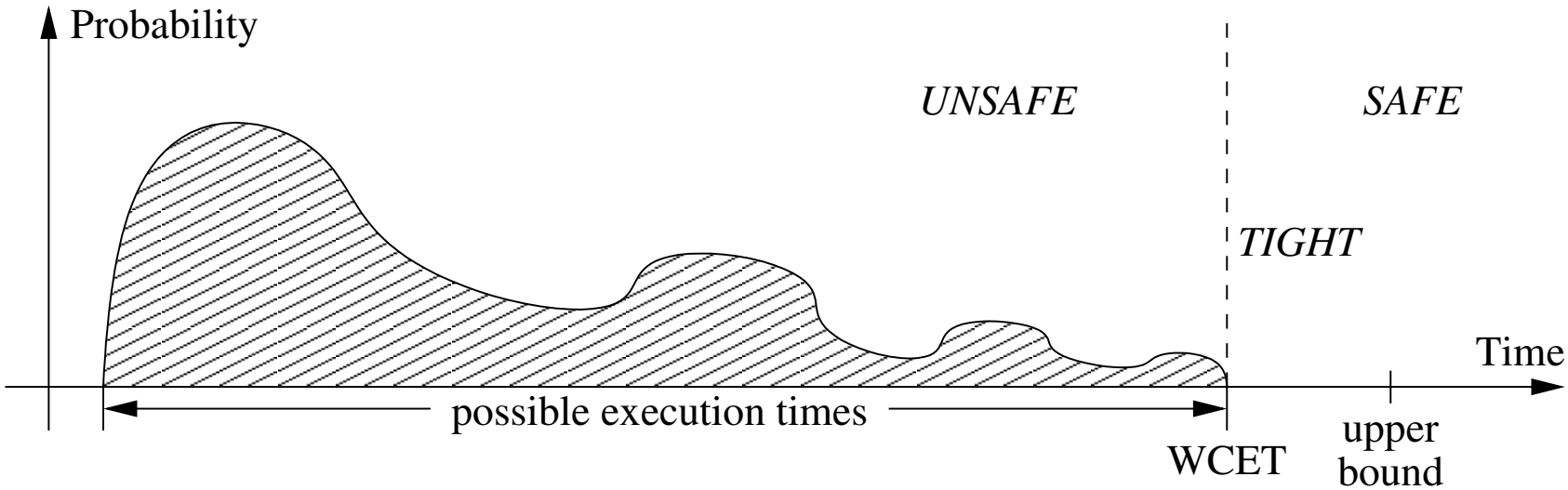
Embedded Software and Real-Time Systems

- Embedded software is everywhere!
- Trend: more software, less hardware
- Easier to upgrade/customize software than hardware
- Systems with embedded software $\supset$ safety-critical systems $\supset$ time-critical/real-time systems (RTSs)
- RTSs must react to events in a timely fashion



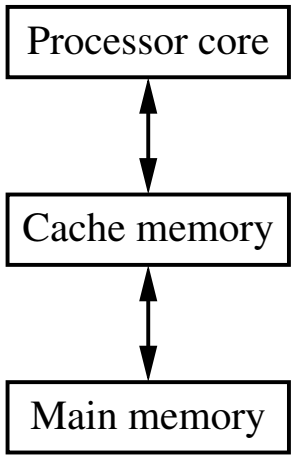
Worst-Case Execution Time Analysis

- Scheduling algorithms for RTSs need WCETs for processes
- Arbitrary inputs must be taken into account
- Measurement-based methods are unsafe
- Tight and safe WCETs yield efficient and reliable RTSs



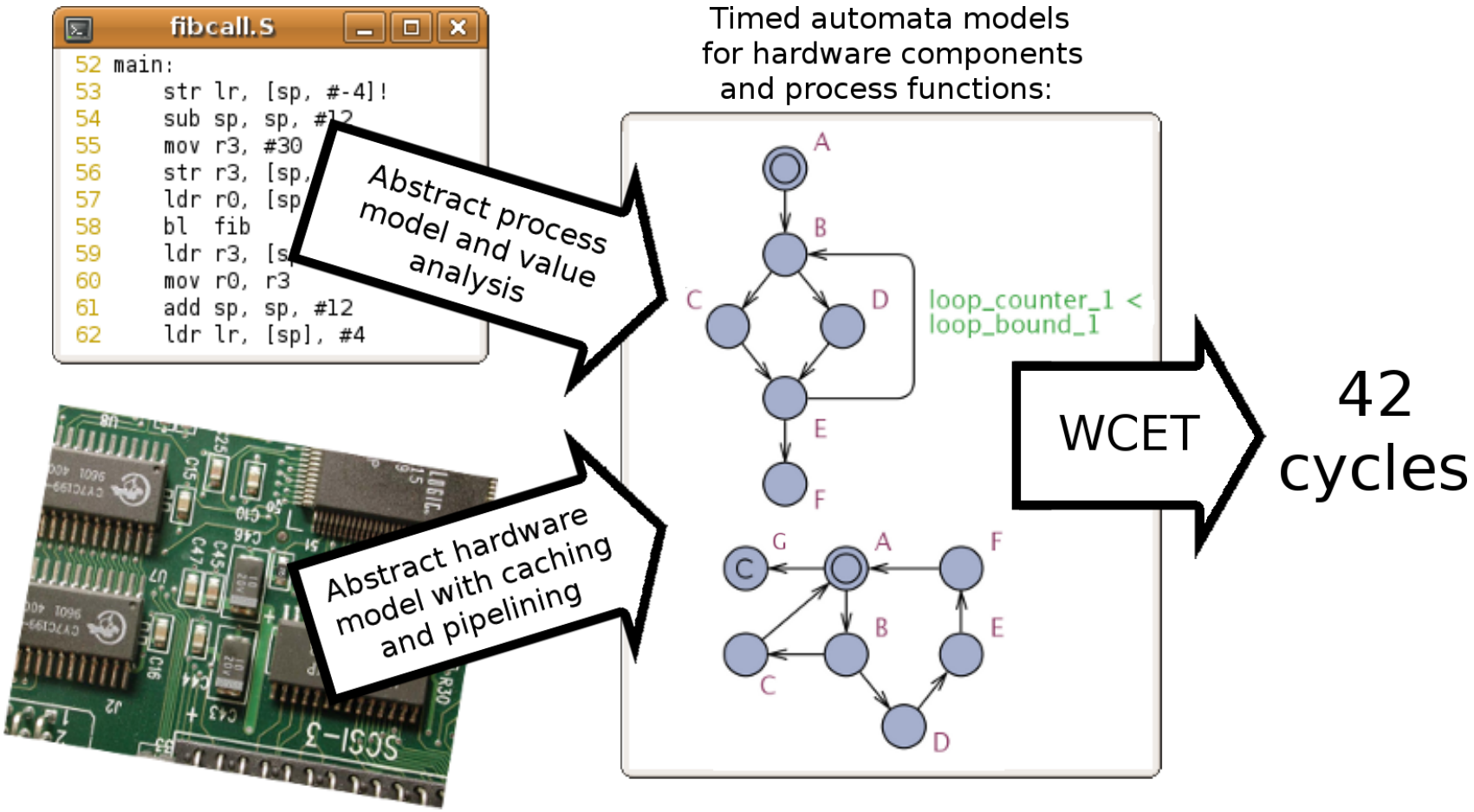
The Complexity of Modern Hardware

- The WCET of a process depends on the hardware platform
- Features of the platform must be taken into account
  - Caching: store frequently used data in a fast memory
  - Pipelining: parallelize the steps involved in executing a process
- There are even more complex techniques out there
  - Branch prediction, out-of-order execution, multicore, ...

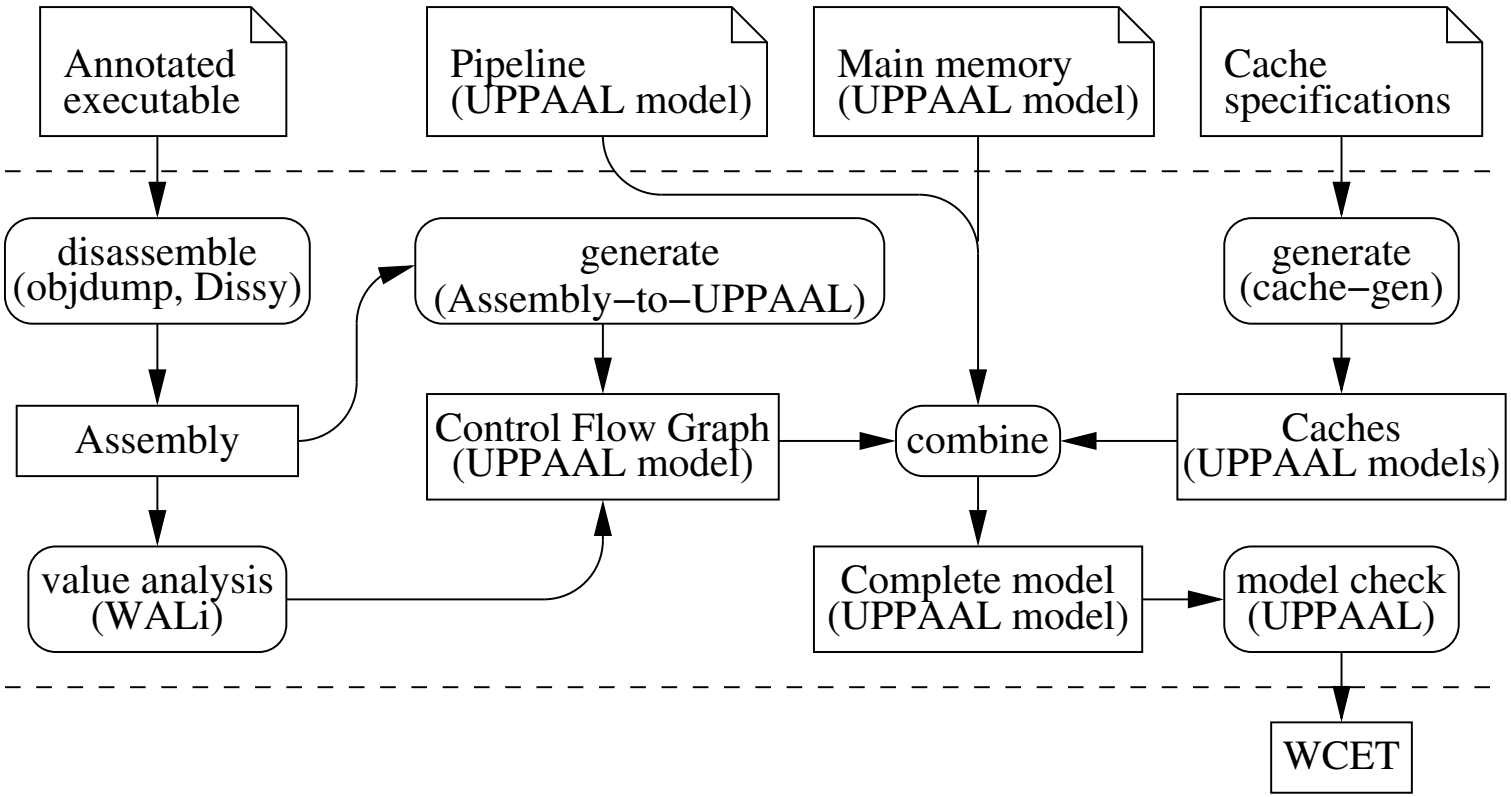


Cycle	Stage 1	Stage 2	Stage 3	Stage 4
1	Instr. 1			
2	Instr. 2	Instr. 1		
3	Instr. 3	Instr. 2	Instr. 1	
4	Instr. 4	Instr. 3	Instr. 2	Instr. 1
5	Instr. 5	Instr. 4	Instr. 3	Instr. 2

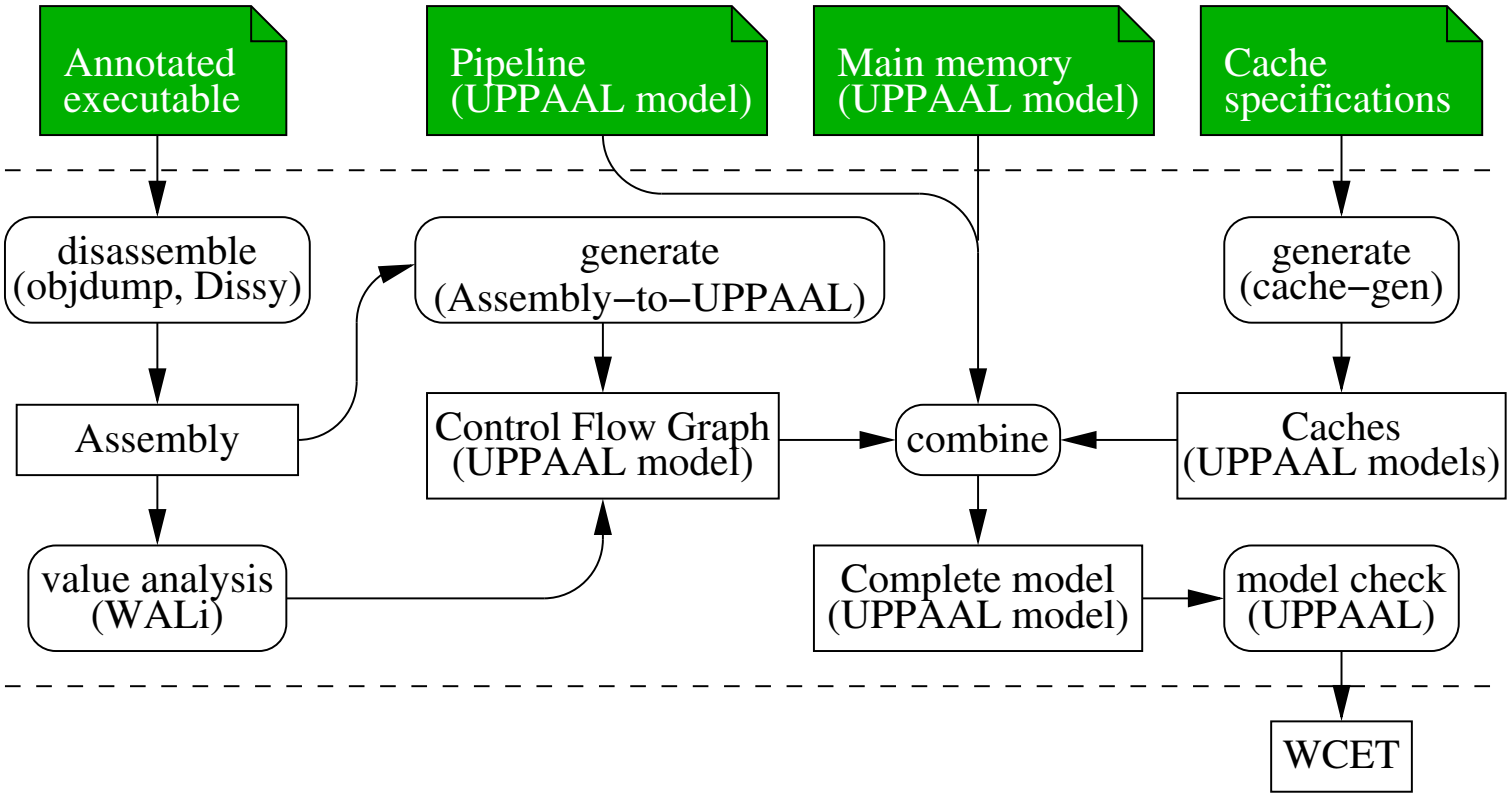
Overview of METAMOC



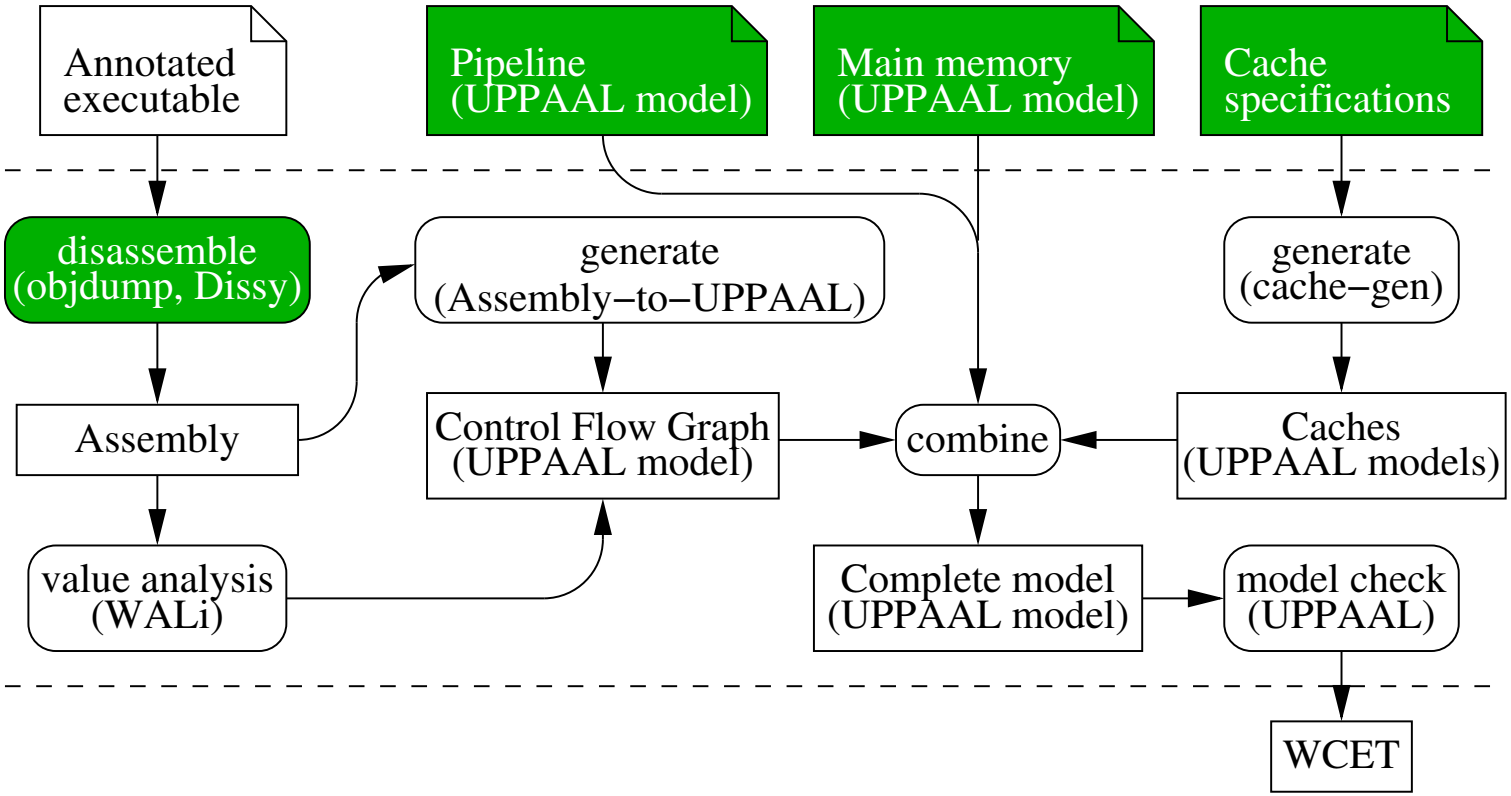
Overview of METAMOC



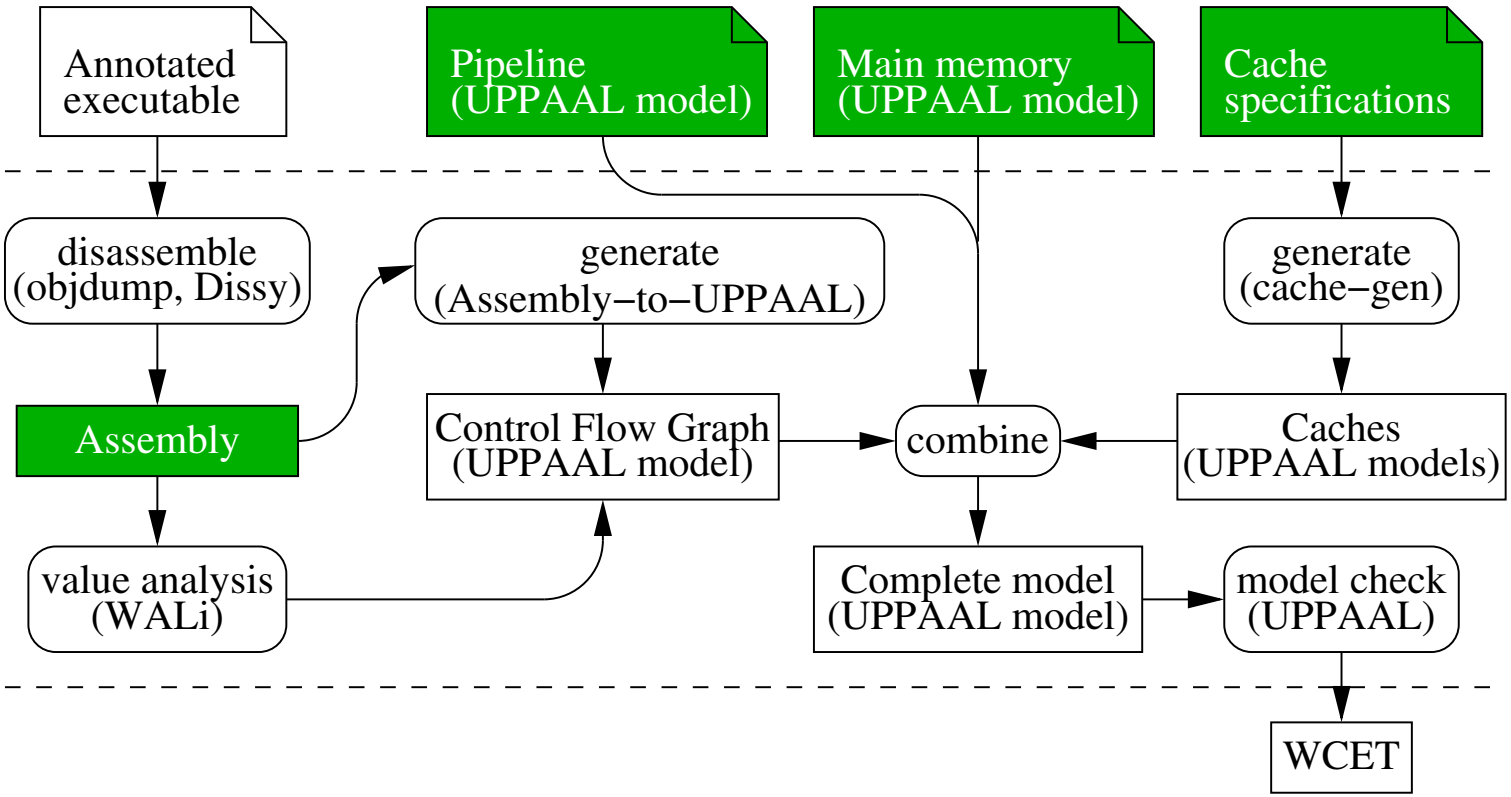
Overview of METAMOC



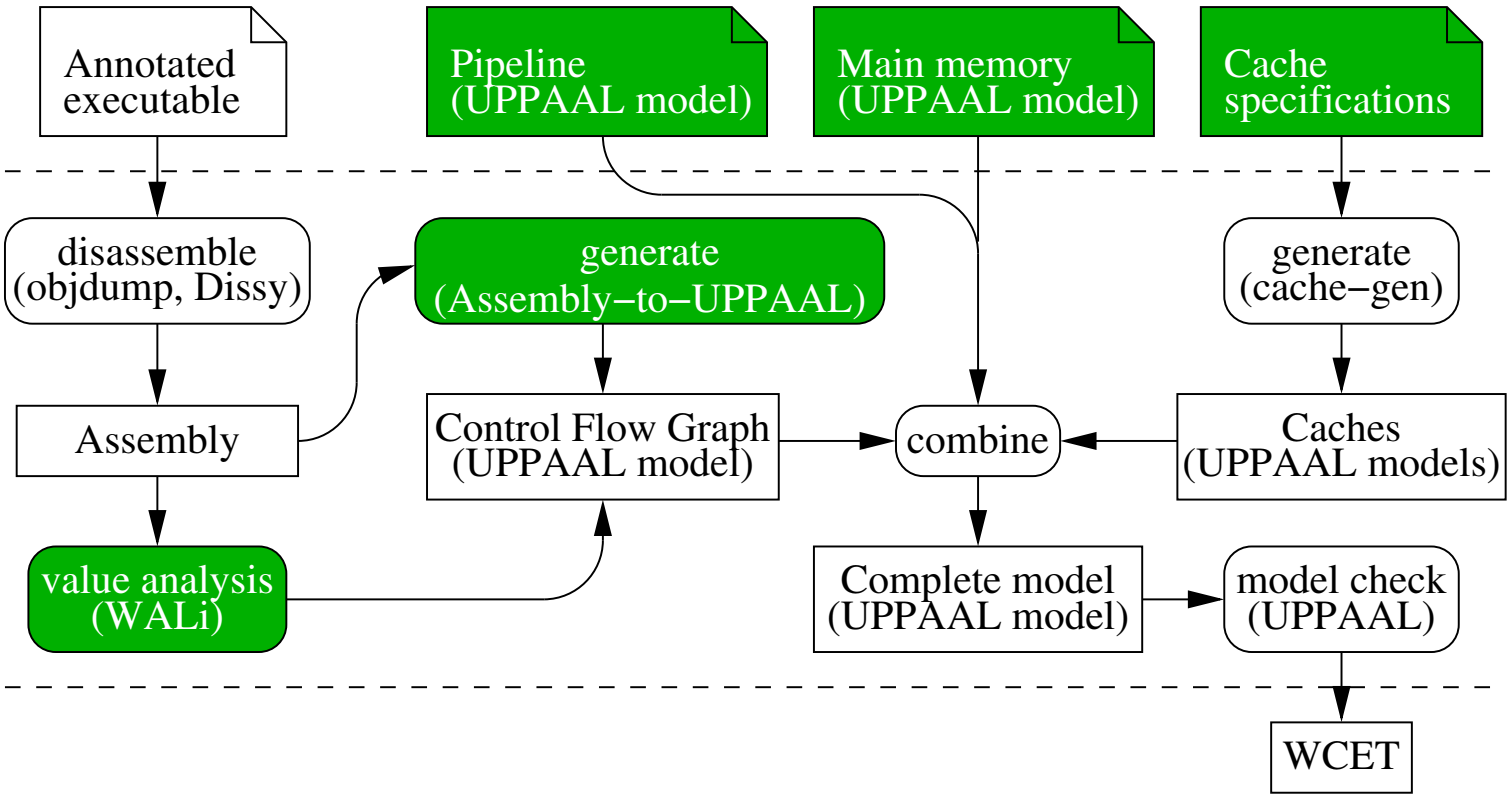
Overview of METAMOC



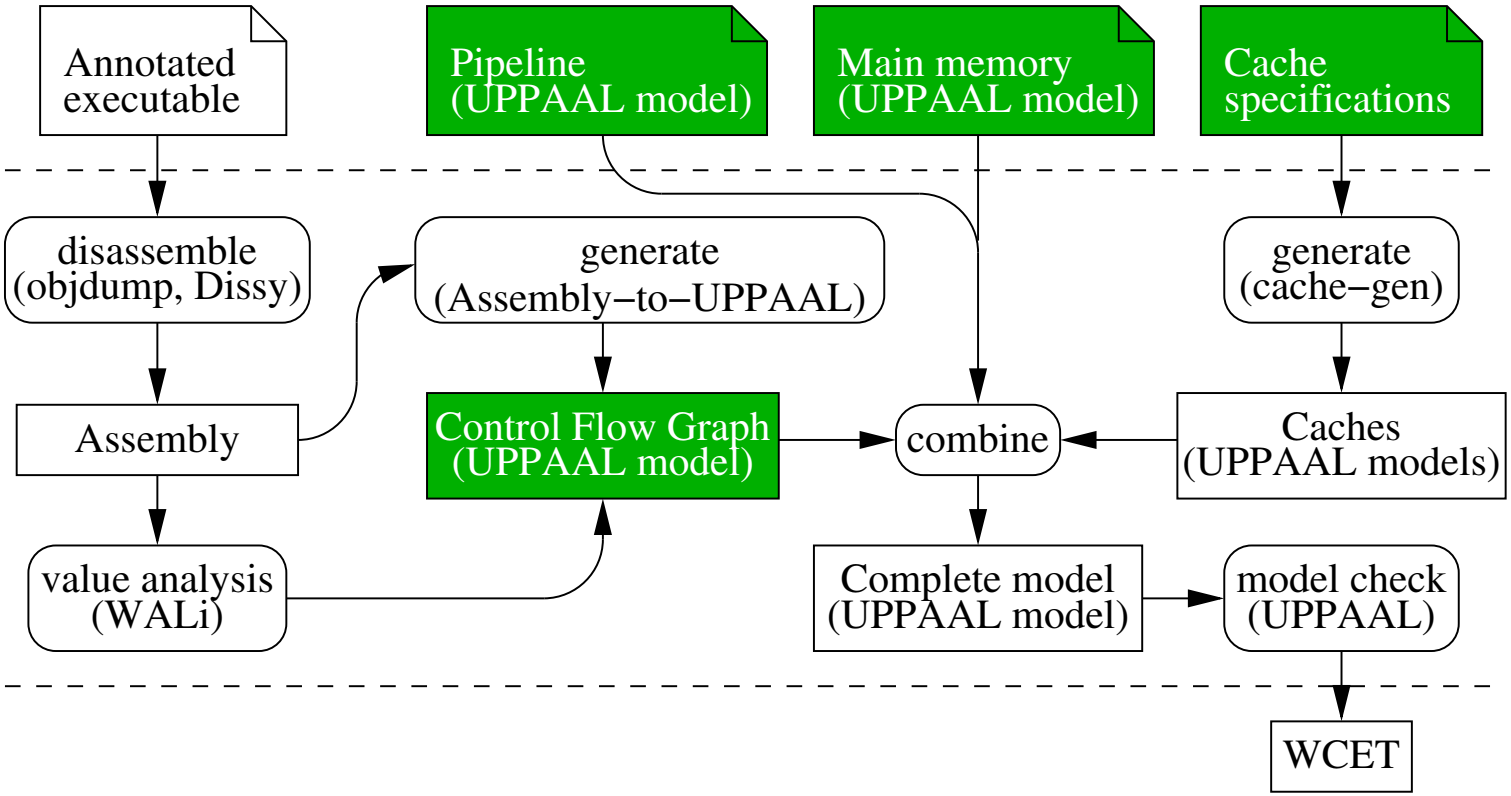
Overview of METAMOC



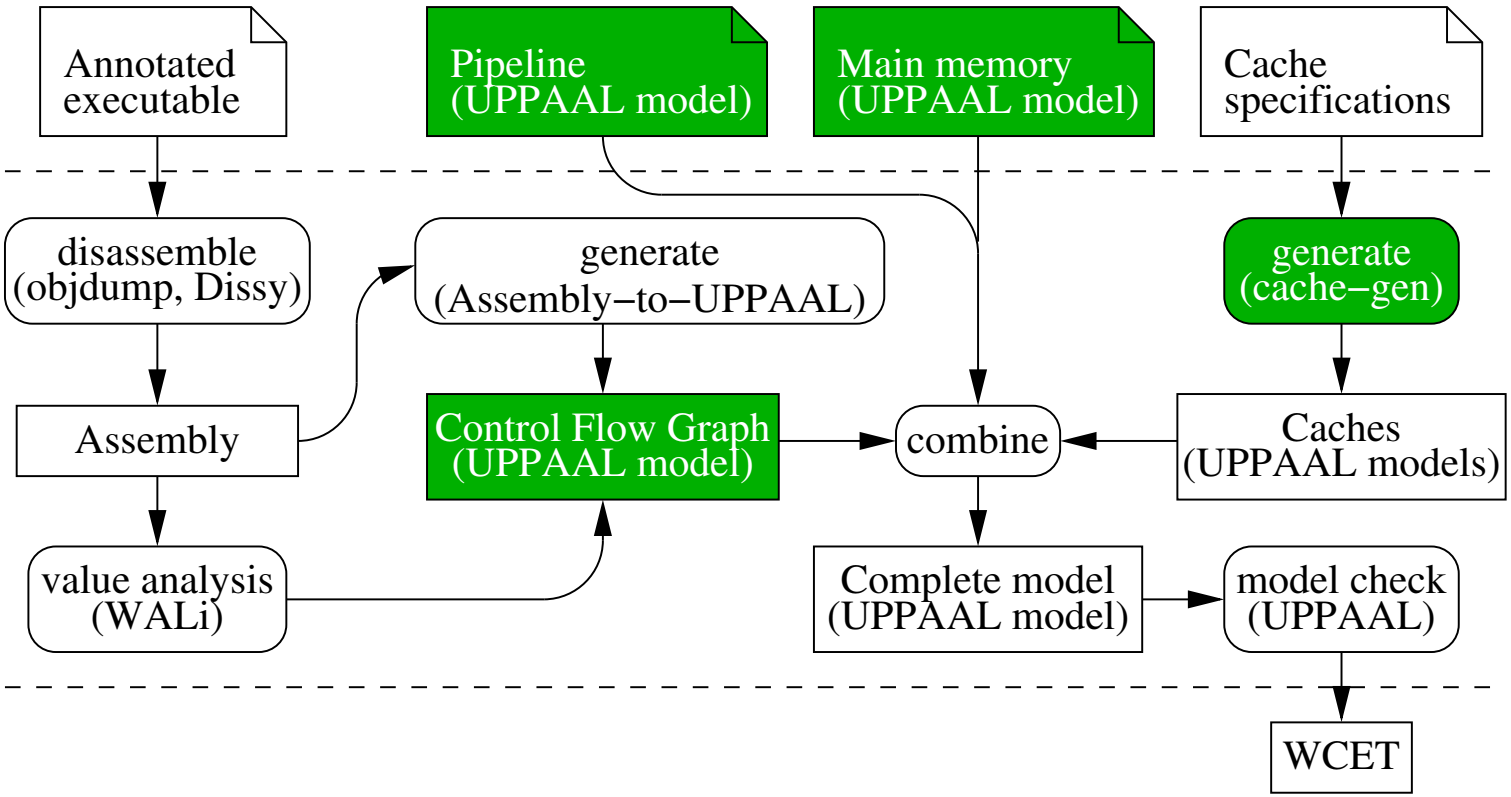
Overview of METAMOC



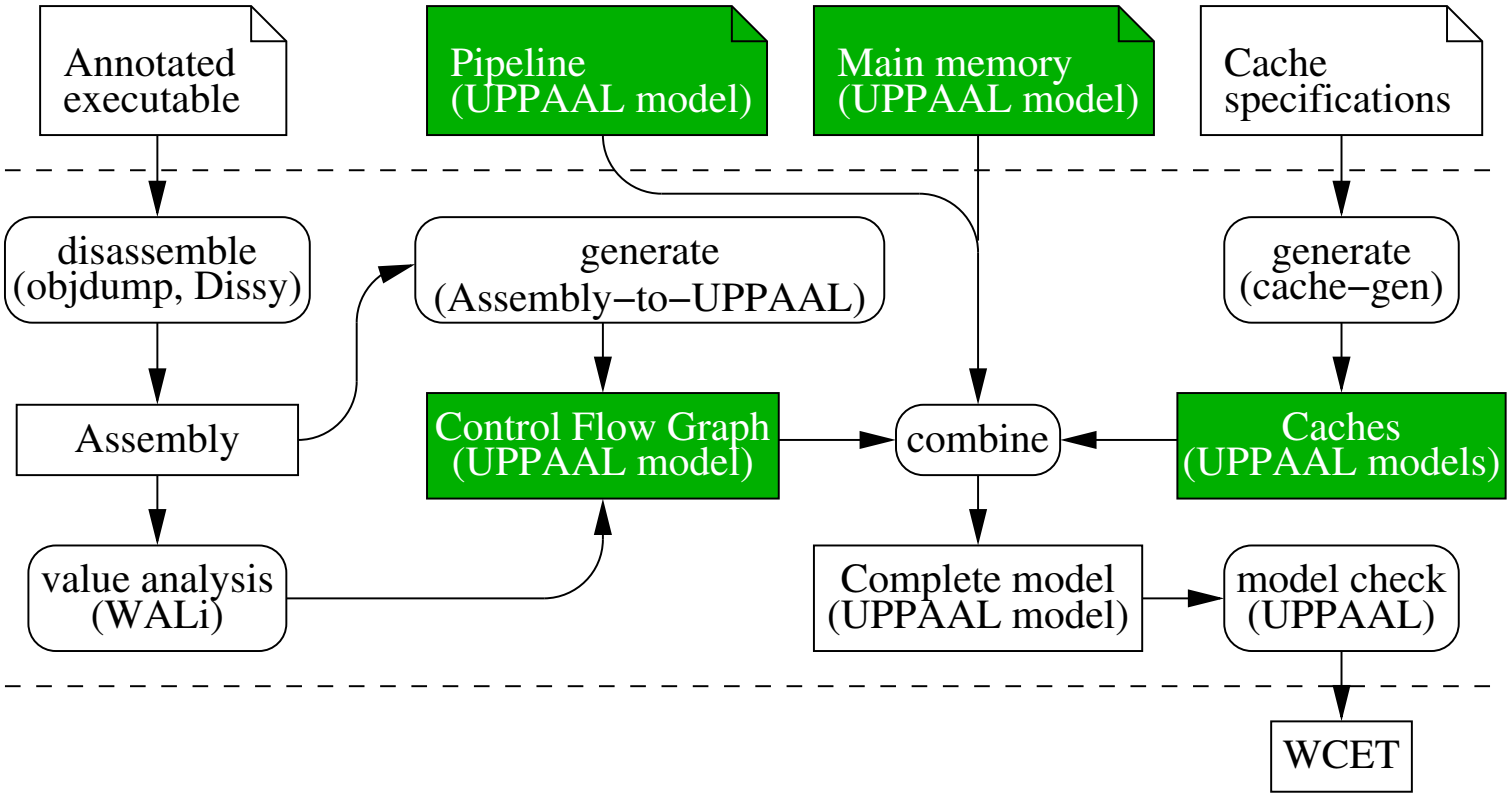
Overview of METAMOC



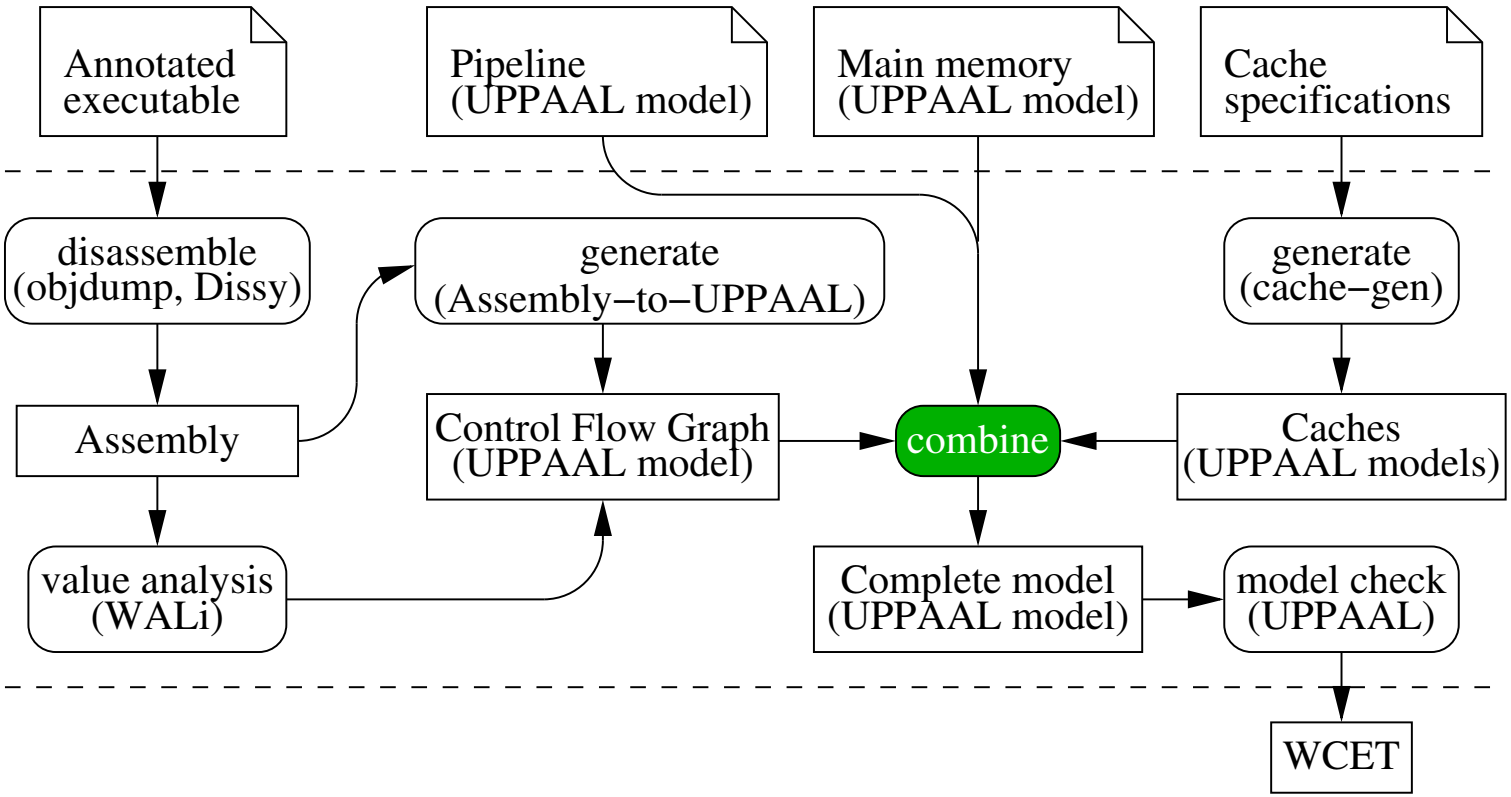
Overview of METAMOC



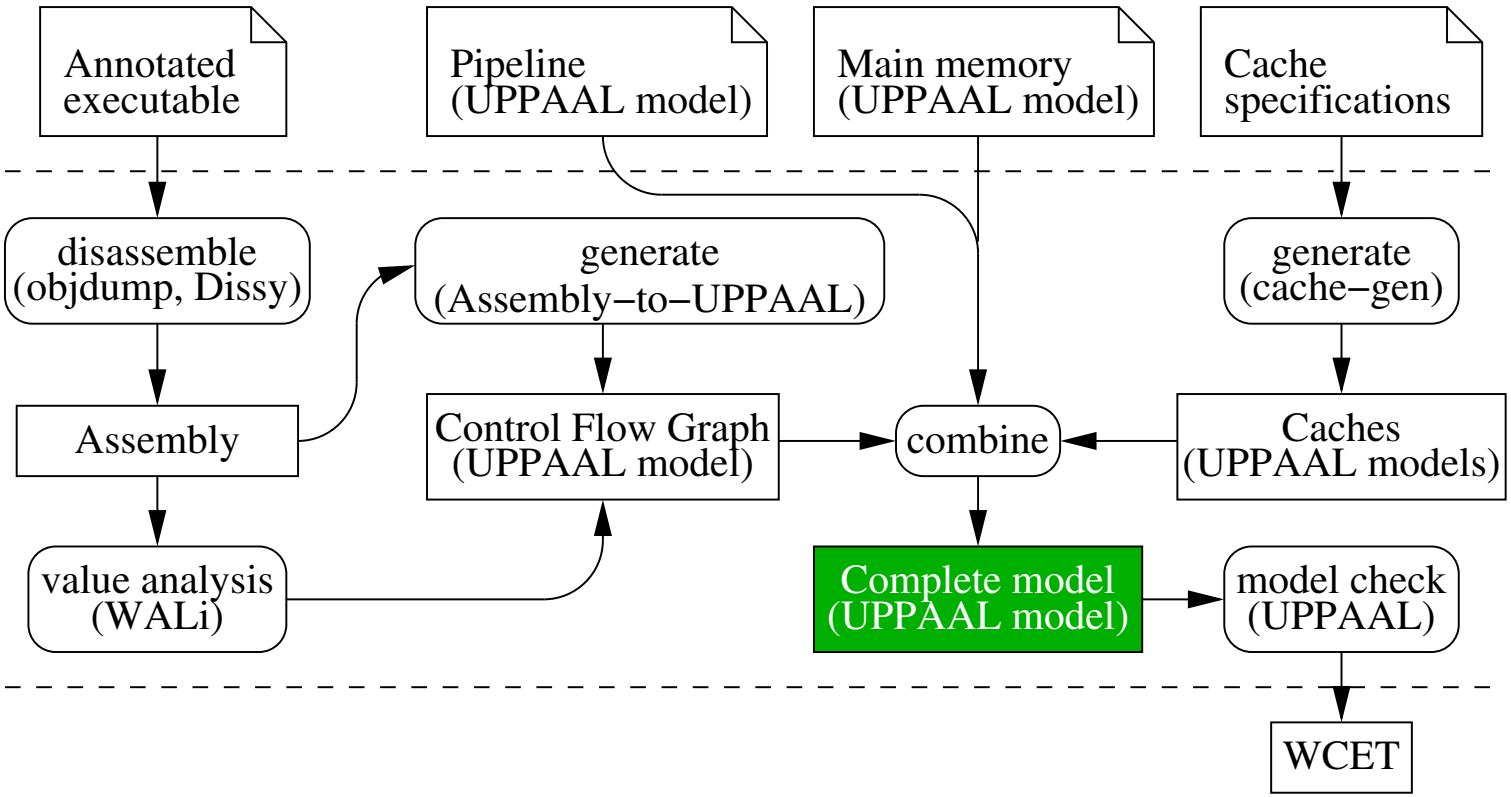
Overview of METAMOC



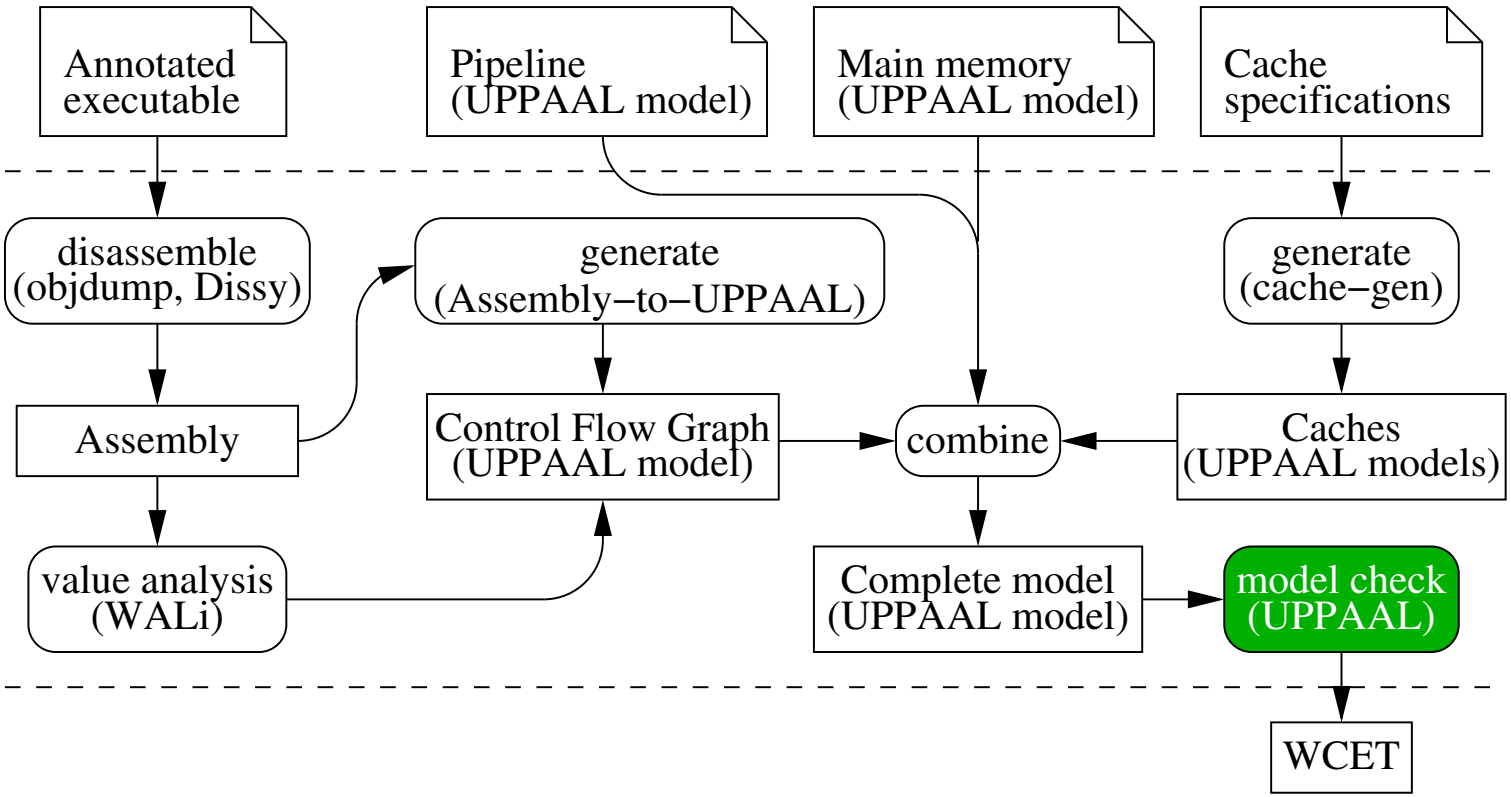
Overview of METAMOC



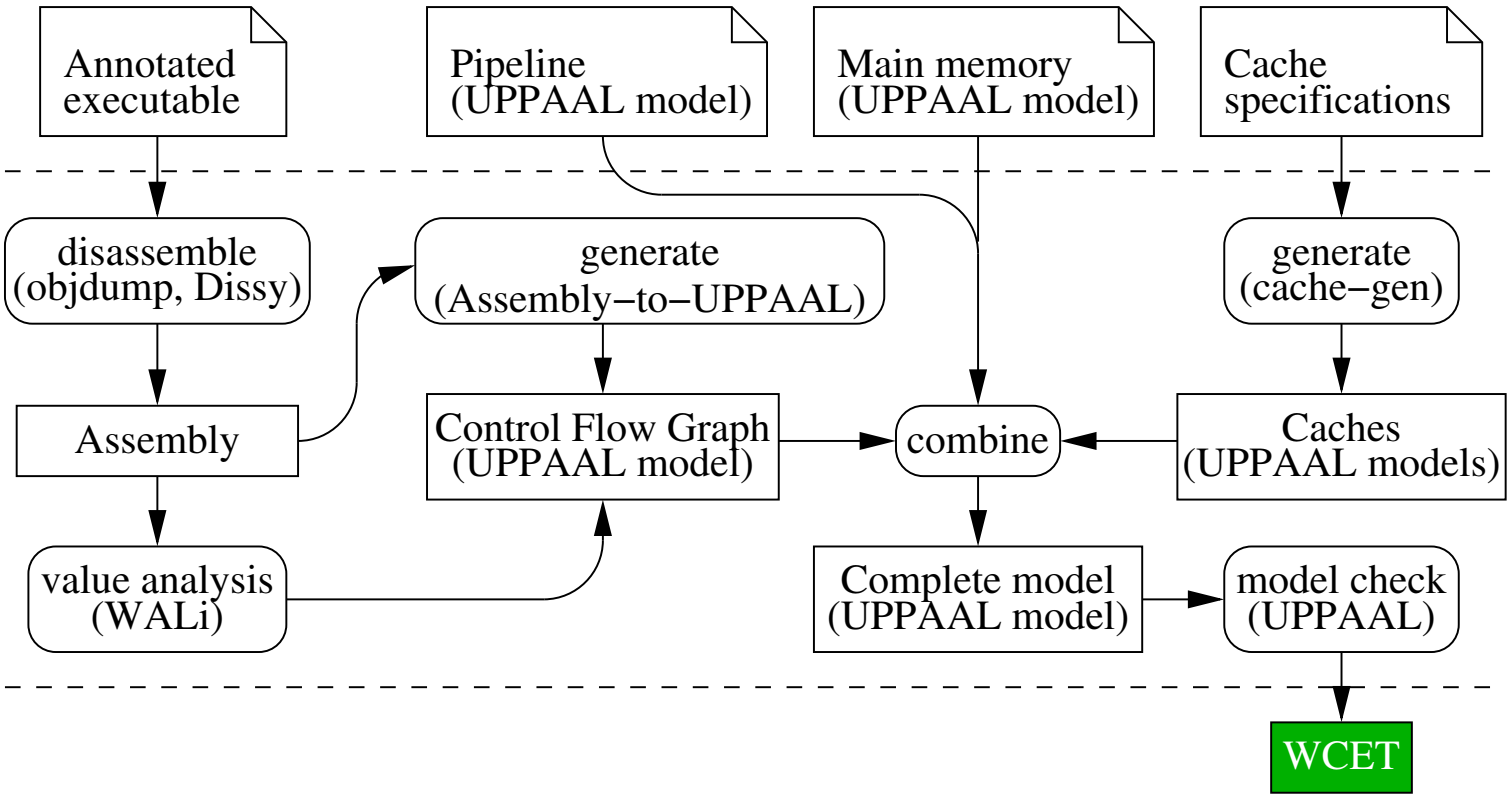
Overview of METAMOC



Overview of METAMOC



Overview of METAMOC

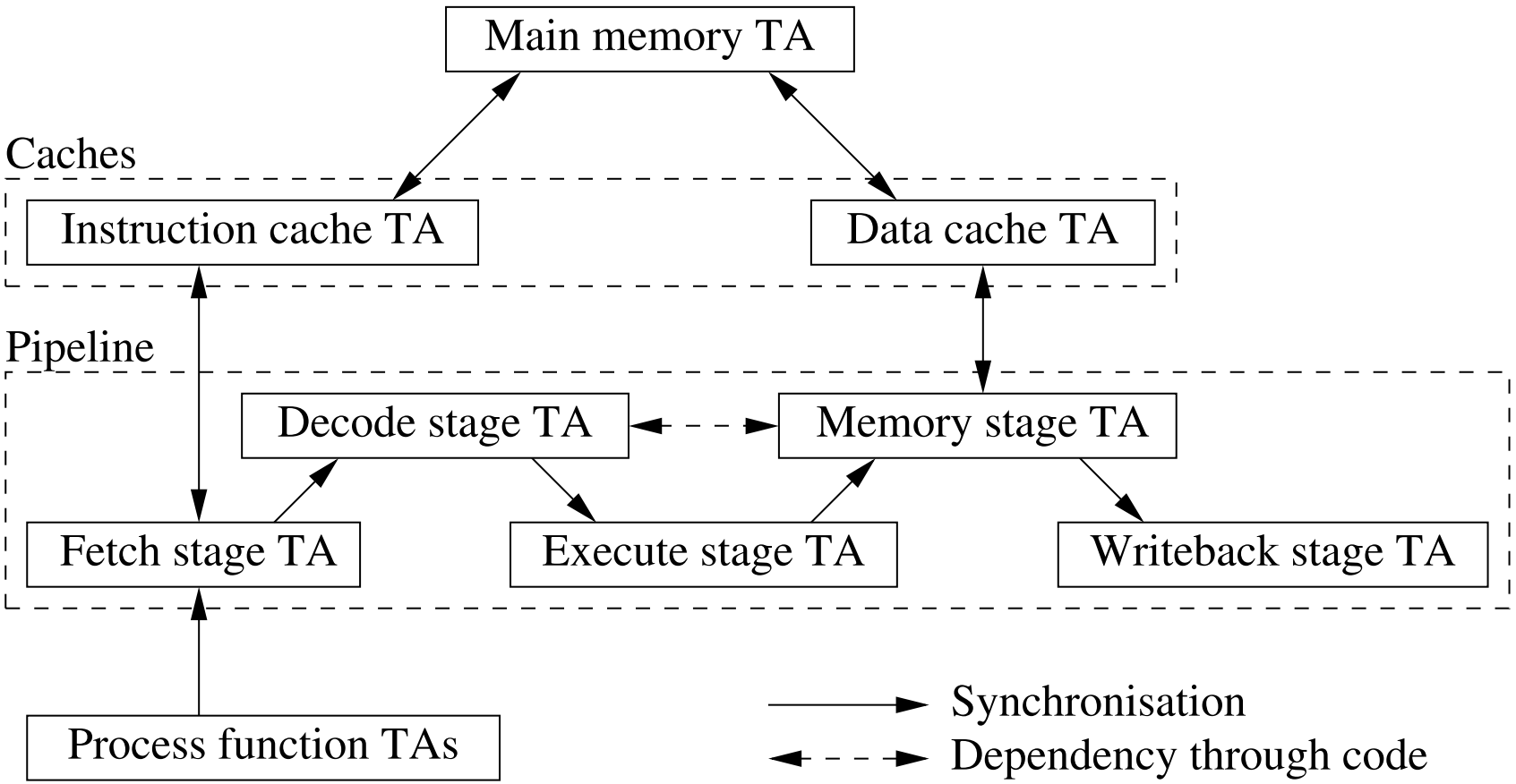


Value Analysis in METAMOC

- Memory addresses needed for cache hit/miss predictions
- Registers used as base and offset for memory accesses
- Overapproximate possible register values
- METAMOC uses Weighted Push-Down Systems (WPDSs) for an inter-procedural, control-flow sensitive value analysis[1]
- Weighted Automata Library (WALi) utilised

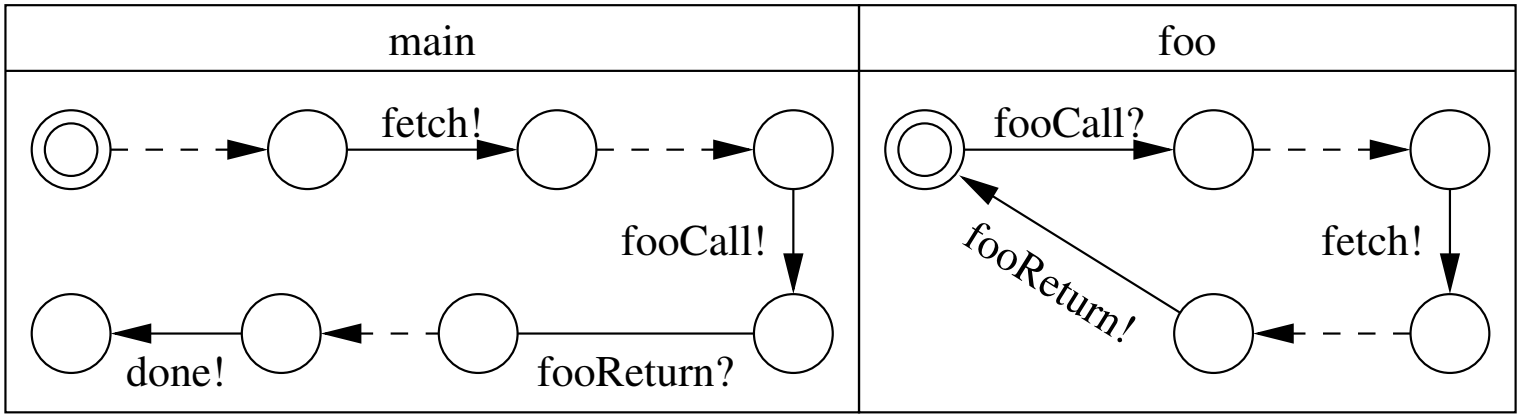
[1] T. Reps, A. Lal, N. Kidd. *Program Analysis using Weighted Push-Down Systems*. In *FSTTCS 2007*, vol. 4855 of *LNCS*, pp. 23–51.

Modelling in METAMOC



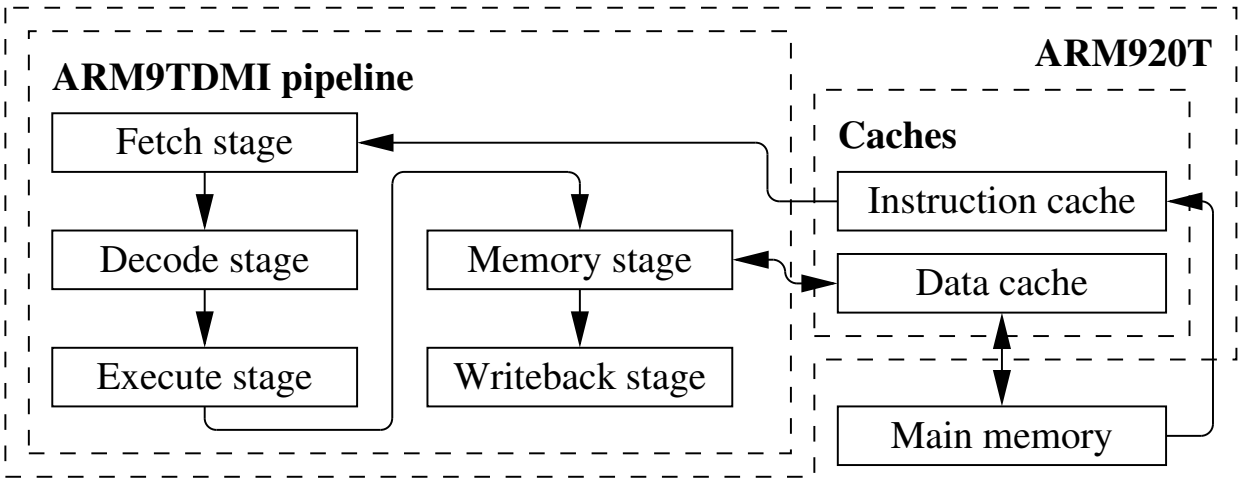
Overview of the ARM9 automata

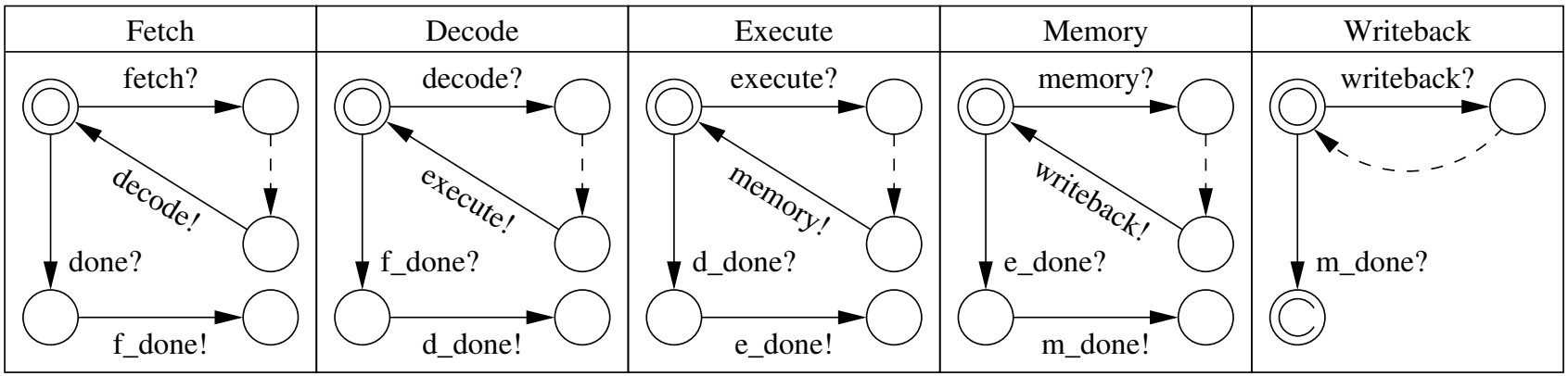
Modelling in METAMOC



Sketch of the function automata for a process

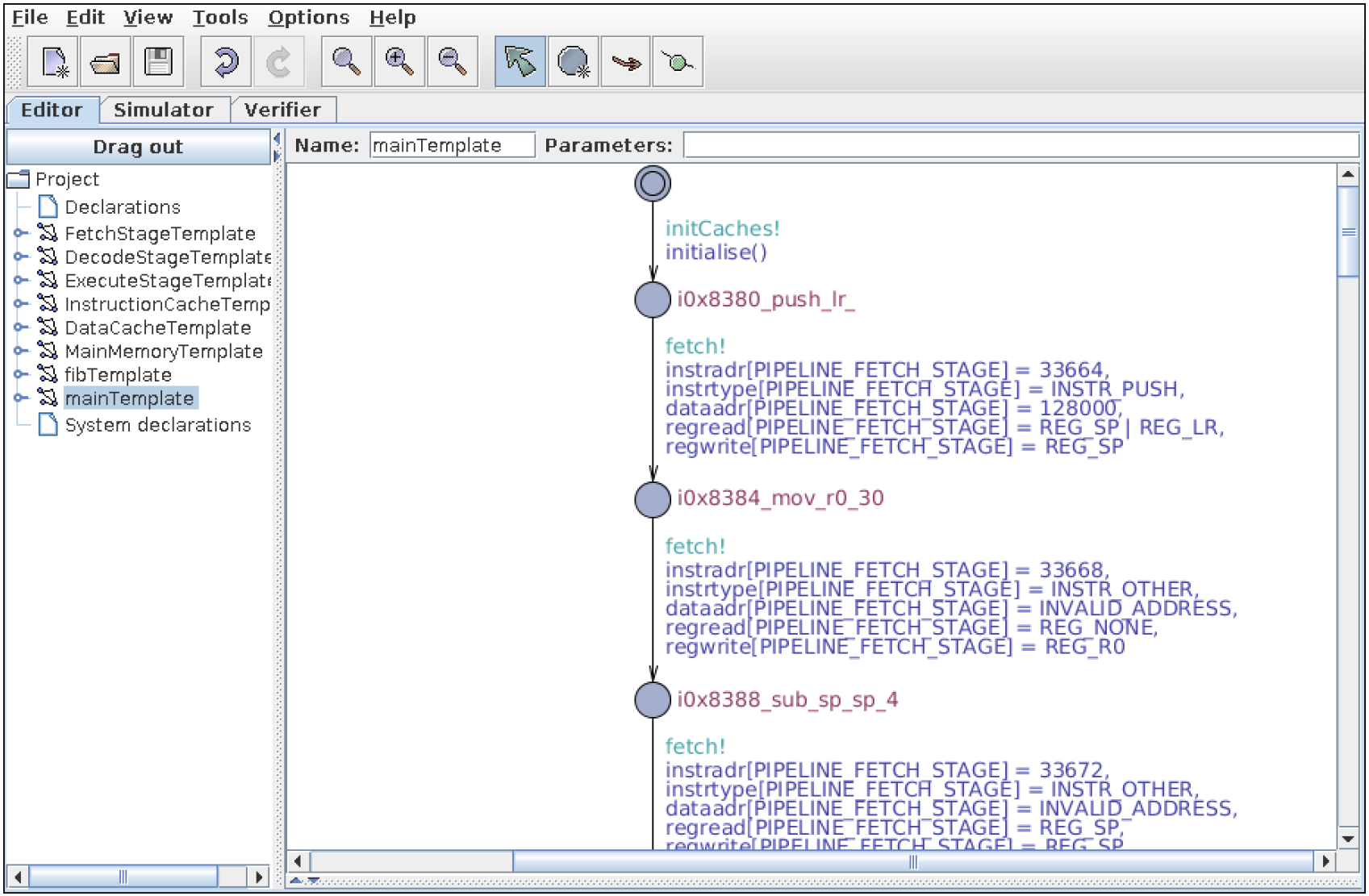
Modelling in METAMOC



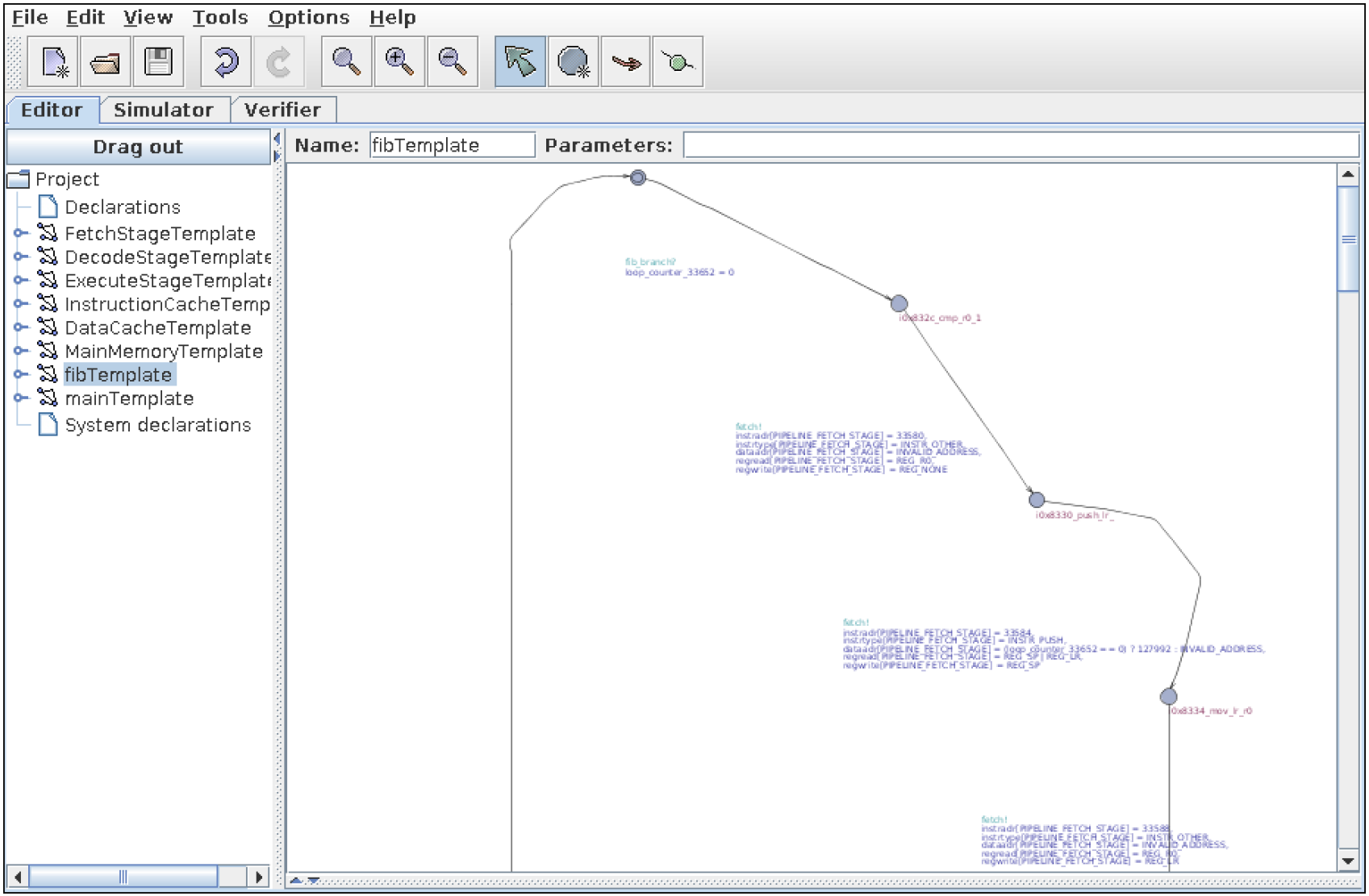


ARM9 overview and sketch of pipeline automata

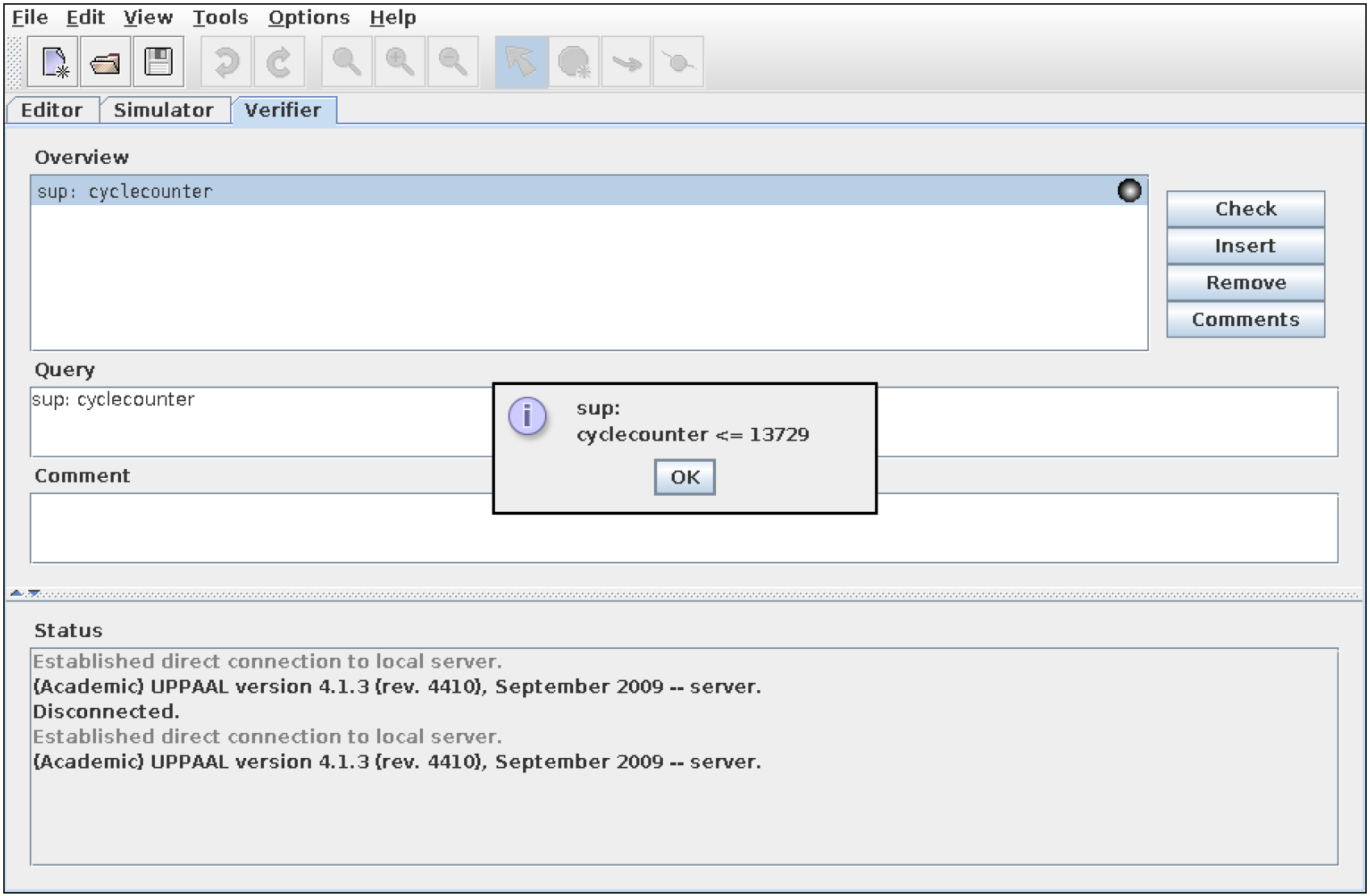
Model Checking using UPPAAL



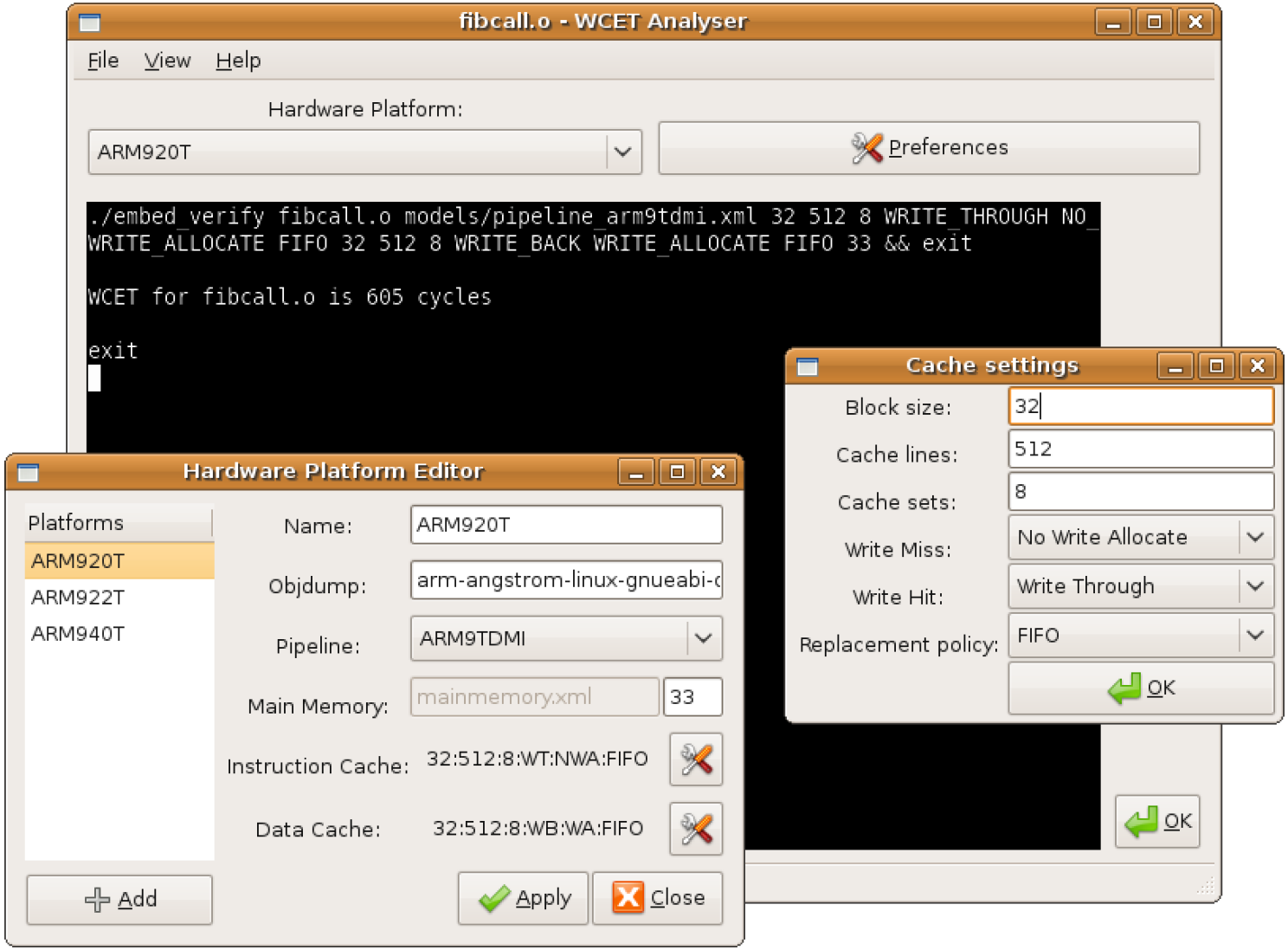
Model Checking using UPPAAL



Model Checking using UPPAAL



Graphical User Interface of METAMOC



Status

- Started out with ARM9 support
  - Five stage pipeline, instruction cache, data cache, simple main memory
- Demonstrated the method convincingly—we thought
- The WCET community: "It's a case study. You haven't demonstrated the method's modularity."
- Now:
  - Support for ARM7, ARM9 and ATMEL AVR 8-bit
  - . . . with modest effort
- Accepted paper for WCET 2010, the 10th Int'l Workshop on Worst-Case Execution-Time Analysis
  - A. E. Dalsgaard, M. C. Olesen, M. Toft, R. R. Hansen and K. G. Larsen. *METAMOC: Modular Execution Time Analysis using Model Checking.*

Experiments

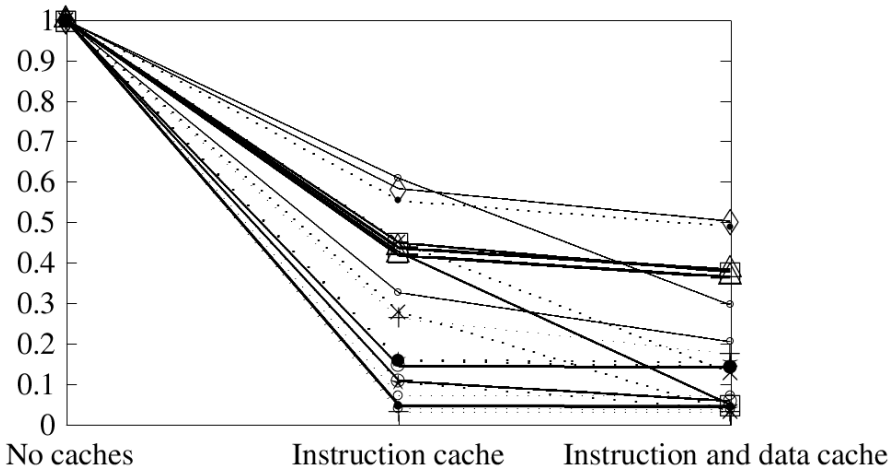
- Evaluation using WCET benchmark programs from Mälardalen Real-Time Research Centre[2]
  - Applicability
  - Performance
- Discarded a number of programs
  - Floating point operations handled by software routines
  - Dynamic jumps
  - Some programs do not compile
- 21 programs for ARM and 19 programs for AVR
- Manually annotated loop bounds

[2] `http://www.mrtc.mdh.se/projects/wcet/benchmarks.html`

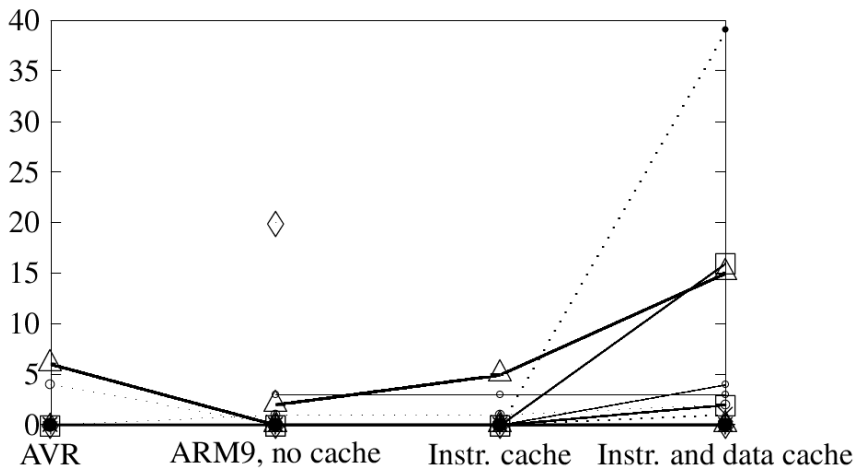
Experiments

ARM9, 21 benchmarks	
Analysable without caches	21
Analysable with instruction cache	20
Unanalysable, state space explosion	1
Analysable with data and instruction cache	20
Unanalysable, state space explosion	1
Manual modification of e.g. data cache size	4

ATMEL AVR 8-bit, 19 benchmarks	
Analysable	16
Unanalysable, state space explosion	3



Relative improvement in WCET for ARM9.



Analysis times in minutes for AVR and ARM9.

Current Work

- The WCET 2010 paper is being revised—it seems FIFO caches give rise to timing anomalies
- Looking for WCET benchmark programs with reference WCETs
- Model checker improvements
  - Distributed model checking
  - Abstract caches using new, lattice-inspired types
- A general, cycle-accurate hardware emulator, turning a hardware description and a program into a WCET

Thank you for your attention!

Questions?

`http://metamoc.martintoft.dk`

`http://martintoft.dk/slides/danes2010.pdf`