

Overture Tool Status & Improvements

Peter Gorm Larsen & Marcel Verhoef

Iteration One (2004)

- Pieter van der Spek (MSc, TU Delft)
 - Build parser and simple pretty printer
 - Experiments on improved error support in parser generator (published as ACM Sigplan Notices)
 - Delivered as Eclipse plug-in
 - *Limited (no) XML support*
 - *Direct manipulation of concrete syntax tree*

Iteration Two (2005)

- Jacob Porsborg Nielsen & Jens Kielsgaard Hansen (MSc, TU Denmark)
 - Re-implemented parser using ANTLR
 - abstract syntax with appropriate Java interfaces
 - XML support for reading / writing AST instances
 - Experimented with Eclipse architecture, many useful suggestions and prototype plug-ins
 - *Hand-coded AST implementation – very error prone*
 - *Many errors in parser implementation*

Meanwhile in 2006

- Address problem of AST maintenance
- Executive decision: we need a **robust** solution...
- ... to support language experiments
- Back To Basics: how can we re-use VDMTools parser and know-how?
- How good are the open source **jflex** and **byaccj** tools?
- Automation is key
- “eat our own dogfood”

AST specification and code examples

```
%directory "d:\projects\ShowTraceParser";  
%package org.overturetool.tracefile.ast;
```

```
TraceFile ::  
  Trace : seq of TraceEvent;
```

```
TraceEvent =  
  ThreadSwapIn | ....  
  DeployObj;
```

```
--  
-- THREADS  
--
```

```
ThreadSwapIn ::  
  -- id of the thread  
  id : nat  
  -- id of the object  
  objref : [nat]  
  -- name of the class  
  clnm : [seq of char]  
  -- id of the CPU  
  cpunm : nat  
  -- swap-in overhead (time units)  
  overhead : nat  
  -- observation time  
  time : nat;
```

```
package org.overturetool.tracefile.ast.itf;
```

```
import jp.co.csk.vdm.toolbox.VDM.*;
```

```
public abstract interface IThreadSwapIn extends ITraceEvent  
{  
    abstract Integer getId() throws CGException;  
    abstract Integer getObjref() throws CGException;  
    abstract Boolean hasObjref() throws CGException;  
    abstract String getClnm() throws CGException;  
    abstract Boolean hasClnm() throws CGException;  
    abstract Integer getCpunm() throws CGException;  
    abstract Integer getOverhead() throws CGException;  
    abstract Integer getTime() throws CGException;  
}
```

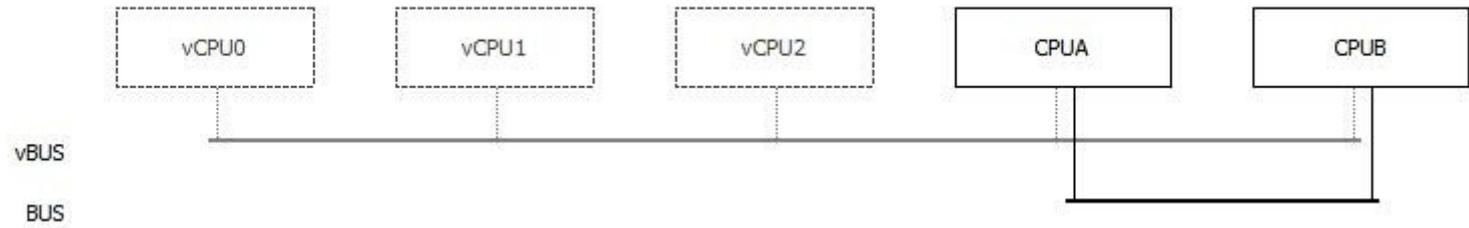
The Proof of the Pudding ...

- Applied this approach to implement Tracefile viewer
- Implemented parser using JFLEX and BYACCJ
- Some “extra” bonuses
 - implemented standard “visitor pattern” support
 - implemented AST attribution “NodeProperty”
 - default visitors for writing XML and VDM-SL values
- Many changes occurred during development
- Turn-around time new parser: just a few hours
- JFLEX / BYACC seem quite robust and FAST
- IMHO: this is the way to go!

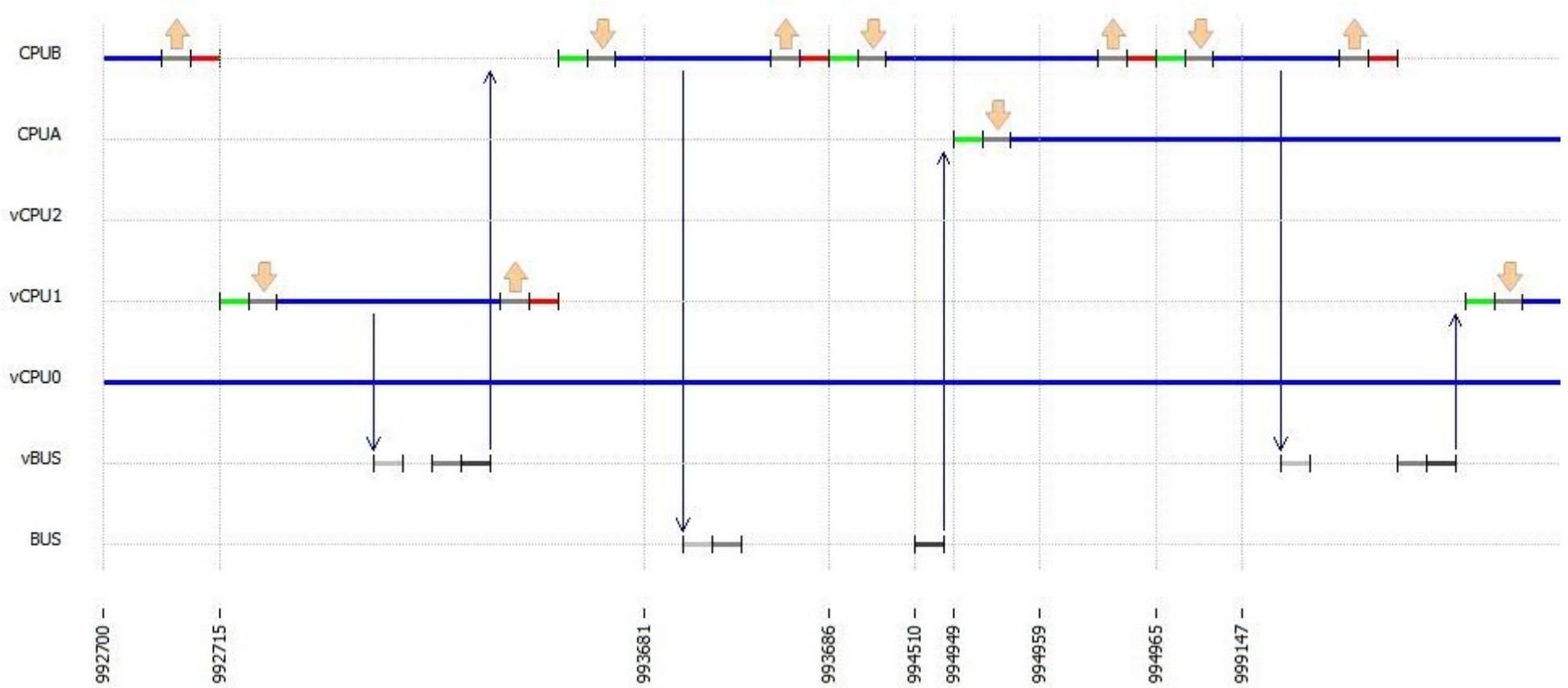
Current and future activities

- Thomas Christensen (Aarhus University, Denmark) started work on Overture static semantics
- Hugo Macedo (Universidade do Minho, Portugal) will start in January on Overture dynamic semantics
- Potential new student from TU Delft
- Marcel Verhoef is working (with Peter Gorm Larsen) on tracefile analysis and visualisation

Tracefile Viewer (1)



Tracefile Viewer (2)



Tracefile Viewer (3)

