Some applications of SAT

What SAT4J users do

Daniel Le Berre

CRIL-CNRS UMR 8188, Université d'Artois, Lens, FRANCE
`leberre@cril.univ-artois.fr`
`http://www.sat4.org/`

DoD SAT Workshop - Baltimore - March 3-5, 2008

Outline

Disclaimer

- Non exhaustive list of SAT-based apps :
  - examples of different users needs
  - wide target applications
- Biased toward Java/SAT4J users
- Not necessarily difficult problems : used in interactive tools
- SAT as a black box for such users : no internal tuning needed, push button technology.

The SAT4J project : *"Java users deserve SAT technology"*

http://www.sat4j.org/

- An open source (EPL/LGPL) library of SAT solvers in Java
- Project started late 2003 as an implementation in Java of the MiniSAT specification.
- Library updated continuously with latest SAT improvements
- See it in action in the SAT competitions (2004, 2005, 2007) and the SAT Race (2006, 2008).
- Can handle several kind of constraints :

clauses	$a \vee b \vee \neg c$
cardinality	$a + b + c + \neg d \geq 3$
pseudo boolean	$3 * a + 2 * b + 2 * c + \neg d \geq 3$

- Constraint Satisfaction Problem (CSP) to SAT support.
- Optimization problems support ([Weighted] MAXSAT).
- Target easy integration in any Java software !
- community support provided by OW2 consortium

What SAT4J users say

- *Warning : Alloy4 defaults to SAT4J since it is pure Java and very reliable. For faster performance, go to Options menu and try another solver like MiniSat.* Felix Chang, MIT. Message appearing when launching Alloy 4.

- *The default SAT Solver used by Forge and JForge is SAT4J, which is pure Java. We highly recommend you use one of the other supported SAT solvers, because they usually exhibit better performance.* Greg Dennis, MIT. Quote from Forge web page.

Use case 1 : the termination competition

http://www.lri.fr/~marche/termination-competition/

- solvers have one minute to prove that a term or string rewriting system terminates, e.g. :

```
INPUT: ( RULES b c -> a b b , b a -> a c b )
ANSWER: NO
Input system R is not terminating since R admits a
looping reduction from bcaaca to aacabacbcaacabbb
with 10 steps.
```

- huge success of the open source SAT solvers MiniSat and SatELite in the SAT 2005 competition
- Aprove, Jambox and Matchbox use them since 2006

SAT in Aprove

http://aprove.informatik.rwth-aachen.de/

- Solving the Lexicographic Path Order (quasi or strict) problem as SAT
- Combine it with Argument Filtering (still as SAT)
- Complete with Recursive Path Order (as SAT)
- Each move to SAT encoding provided orders of magnitude speedup compared to previous techniques
- Aprove, written in Java, used SAT4J for its builtin Tseitin translation !, and Minisat/SatElite as backend SAT solver !
- Remark 1 : CNF format is not user friendly !

More information about Aprove and SAT

- C. Fuhs, J. Giesl, A. Middeldorp, P. Schneider-Kamp, R. Thiemann, and H. Zankl SAT Solving for Termination Analysis with Polynomial Interpretations In Proceedings of SAT '07, Lisbon, Portugal, 2007

- M. Codish, P. Schneider-Kamp, V. Lagoon, R. Thiemann, and J. Giesl SAT Solving for Argument Filterings. In Proceedings of LPAR '06, Phnom Penh, Cambodia, LNAI 4246, pages 30-44, 2006

Use case 2 : Genetic Network Analyzer (bioinformatic)

`http://www.genostar.fr/`

- Analysis of Genetic Regulatory Networks
- GRN = interaction between Genes, proteins, RNA and molecules.
- Abstraction based on state transition graphs
- Objective : finding all steady states
- How : one steady state = one model of a CNF
- Approach : Efficient CNF encoding (not exponential in the number of genes but in the number of regulators of the genes)
- Benefit :
  - real bacterial GRN solved in a few seconds
  - orders of magnitude improvement on previous methods based on CSP on random networks.
- Available in Genostar product since october 2007
- Remark 2 : solving SAT decision problem is not sufficient

More information about GNA and SAT

- H. de Jong, M. Page (2008), Search for steady states of piecewise-linear differential equation models of genetic regulatory networks, ACM/IEEE Transactions on Computational Biology and Bioinformatics, in press.

Use case 3 : Verifying the Executable UML Metamodel

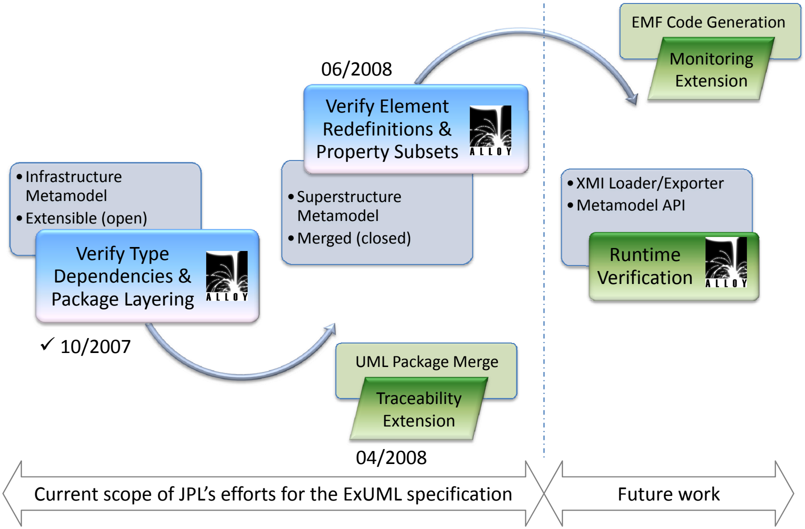
Context of the problem

- Object Management Group (OMG) initiative
- Model Driven Architecture context
- Semantics of a Foundational Subset for Executable UML
- Objective : Enable tool-support for constructing, verifying, translating & executing computationally-complete models
- Check the ExUML metamodel for compliance w.r.t. OMG's language architecture principles
- Package merge is an important step for the well-formed construction of a metamodel for ExUML
- Very important for embedded and/or mission-critical systems
- Approach :
  1. Extract information from the ExUML metamodel with OCL queries
  2. Verify the extracted information using Alloy4
  3. Visualize positive/negative results with Alloy4Eclipse

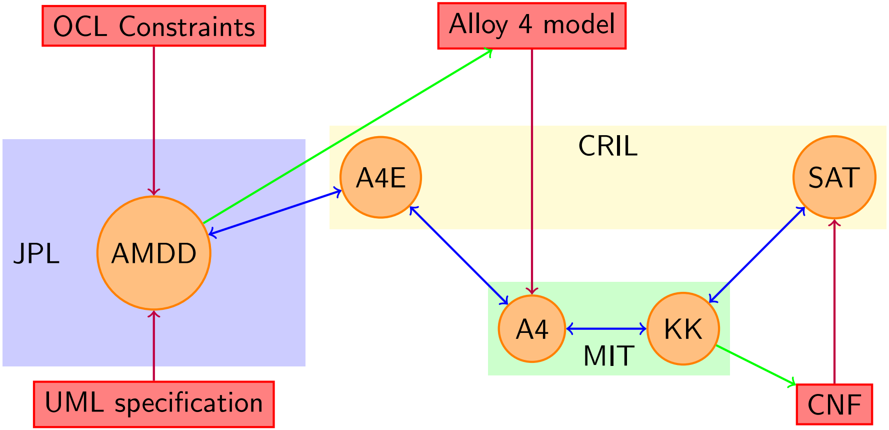
JPL@NASA : Model Operability project

AlloyModelDrivenDevelopment by Nicolas Rouquette

Better Model Driven Development w/ Alloy



The long path from ExUML to SAT



AMDD = AlloyModelDrivenDevelopment,
A4E = Alloy 4 Eclipse, A4 = Alloy 4, KK=Kodkod

Key features

- JPL environment based on Rational Software Architect (Eclipse)
- Alloy4Eclipse provides seamless integration of Alloy4 in the Eclipse environment
- Alloy4Eclipse augments Alloy4 visualization with a basic mapping to elements of the analyzed model
- Alloy 4 checks A4 models and provides visual feedback of models or counterexamples.
- KodKod contains all the machinery for efficient relational model to SAT translation and interpretation (symmetry breaking predicates, unsat core in the original model, etc) and multiple SAT solvers support (berkmin, minisat, zchaff, SAT4J, ...).
- Remark 3 : Alloy 4 separates SAT translation from Alloy 4 semantics. Kodkod can be reused in other tools : Forge

For further information

Kodkod	`http://web.mit.edu/~emina/www/kodkod.html`
Alloy	`http://alloy.mit.edu` Tutorial by Greg Dennis this afternoon !
Forge	`http://sdg.csail.mit.edu/forge/` Bounded Program Verification using Kodkod
Alloy4Eclipse	`http://alloy4eclipse.googlecode.com/`
AlloyModelDrivenDevelopment	to be presented at EclipseCon 2008 conference (17th-20th March). Report and tool should be publicly available soon.

Use case 4 : Eclipse provisioning
Joint work with Anne Parrain (CRIL) and Pascal Rapicault (IBM)

- Eclipse is an open platform for building Rich Client Applications.
- Each functionality is brought by a bundle.
- Each bundle has dependencies (needs, conflicts) with other bundles.
- Bundles can have several versions
- Most recent available bundles should be preferred
- Problem translated into a binate covering problem (CNF + optimization function)
- Manage those dependencies using a Pseudo Boolean solver
- Based on the EDOS project (SAT) and the OPIUM (PB) results for Linux package management.
- Should ship in june with next Eclipse release : 3.4

Eclipse provisioning preliminary results

1structure !20structure !5

Name	Answer	#V	#C	Opt. Sol.	Total
Eclipse IUs from a repository containing all of Ganymede M4					
CDT	SAT	109	770	0.24	1.75
RAP	SAT	128	985	0.33	4.14
TPTP	SAT	197	1338	0.40	3.76
WTP	UNSAT	264	2028	-	0.44
Eclipse release					
SDK 3.3	SAT	376	2231	0.56	1.36

- Can easily find the optimal solution
- Need more time to prove it is optimal !
- Ideal case : nothing installed in the computer. Need to take into account installed bundles too !
- More information at `http://wiki.eclipse.org/Equinox_p2`

Conclusion

- Numerous other applications : requirements engineering, software product lines/feature programming, C static code analyzer, ...
- SAT is a mature technology
- Not only used by power users, but also "normal" programmers
- SAT technology is being used as a lightweight generic search framework
- Improvements needed :
  - Dimacs format is not suitable for average user : CNF, readability
  - Need a more user friendly modeling language, or front end (i.e. MX, Kodkod, BoolVar, etc.)
  - Answering SAT is often not sufficient : enumerating them, proof logging, unsat core, etc. also needed ! One SAT solver usually does not provide all those features

Gartner Hype Cycle for SAT : where are we up to ?

http ://www.gartner.com/

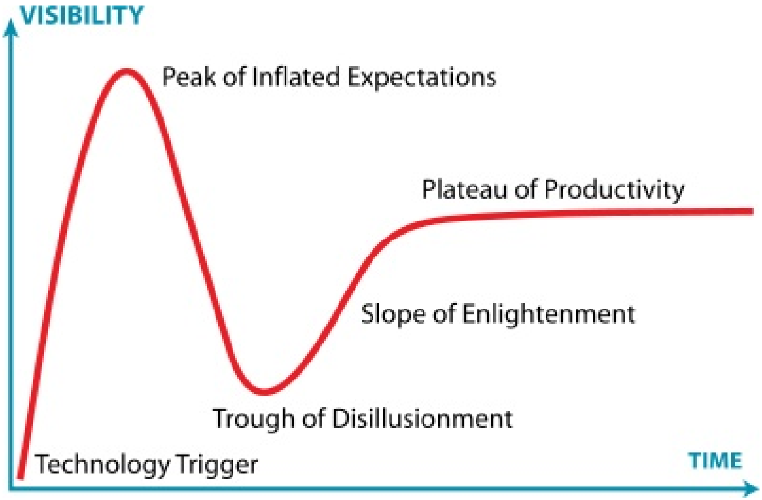


Image from wikipedia

Thanks for your attention

Questions ?

OpenOME
http://www.cs.toronto.edu/km/openome/

OpenOME	an Eclipse plugin for requirements engineering.
goal model	to connect the user's high level requirements with the system's low level configuration items
preferences	between goals : one goal is more important than another
expectations	a goal needs to be satisfied to a certain degree

- Top-down reasoning propagates the high level goals downward to obtain the minimal number of low level goals.
- Top-down reasoning done with SAT4J

AHEAD tool suite

http://www.cs.utexas.edu/users/schwartz/ATS.html

Product line	an aggregation of components having features
Constraint	Not all features are compatible
Safe Composition	Avoiding type errors in the composed code.
AHEAD	theory of software synthesis including feature modeling.

SAT used to :

- debug feature models
- perform safe composition

AHEAD tool suite

http://www.cs.utexas.edu/users/schwartz/ATS.html

Product line	an aggregation of components having features
Constraint	Not all features are compatible
Safe Composition	Avoiding type errors in the composed code.
AHEAD	theory of software synthesis including feature modeling.

"Further, the performance of using SAT solvers to prove theorems was encouraging : non-trivial product-lines of programs of respectable size [...] could be analyzed and verified in less than 30s."

Don Batory and Sahil Thaker, Safe Composition of Product Lines

Alloy 4 Analyzer

http://alloy.mit.edu/

- Structural modelling language based on first-order logic
  - quantifiers
  - higher arity relations
  - polymorphism
  - subtyping
  - ...
- Provides fully automatic simulation and checking
- Uses Kodkod SAT-based model finder
- Alloy can be seen as a subset of Z, ASCII notation
- Alloy is similar to OCL, but fully declarative
- Mature technology (started in 1997)

Alloy4Eclipse

http://alloy4eclipse.googlecode.com/

Aims :

- To provide JDT-like features to Alloy 4 model designers
  - First class Alloy 4 editor (code completion, syntax coloring, refactoring, etc).
  - Compile on save approach.
  - Eclipse like (Outline view, console, preferences...)
- To facilitate the integration of Alloy 4 with other Eclipse plugins.
  - Model files management the Eclipse way
  - SWT/SWING integration of Alloy 4 graphical views.

Development :

- Started in January 2007
- By two 4th year University Students (Université d'Artois, Lens, France).
- Two new students working on it in 2008