CS846 Paper Review Form - Winter 2012

Reviewer: David Dietrich

Paper Title: Alloy: A Lightweight Object Modeling Notation

Author(s): Daniel Jackson

1) Is the paper technically correct?
 [x] Yes
 [ ] Mostly (minor flaws, but mostly solid)
 [ ] No

2) Originality
 [ ] Very good (very novel, trailblazing work)
 [x] Good
 [ ] Marginal (very incremental)
 [ ] Poor (little or nothing that is new)

3) Technical Depth
 [ ] Very good (comparable to best conference papers)
 [x] Good (comparable to typical conference papers)
 [ ] Marginal depth
 [ ] Little or no depth

4) Impact/Significance
 [ ] Very significant
 [x] Significant
 [ ] Marginal significance.
 [ ] Little or no significance.

5) Presentation
 [ ] Very well written
 [x] Generally well written
 [ ] Readable
 [ ] Needs considerable work
 [ ] Unacceptably bad

6) Overall Rating
 [x] Strong accept (award quality)
 [ ] Accept (high quality - would argue for acceptance)
 [ ] Weak Accept (borderline, but lean towards acceptance)
 [ ] Weak Reject (not sure why this paper was published)

7) Summary of the paper's main contribution and rationale

for your recommendation. (1-2 paragraphs)

In the paper Alloy: A Lightweight Object Modeling Notation the author describes
a new modeling language for semantic analysis of consistency and consequences of
a program. What's important to this is that the analysis can be fully automated.
This is important as in MBSE one of the first steps after gathering requirements
is generally to create models that determine if the requirements are correct and
complete. The automated analysis of Alloy lends itself to the checking of
requirements, thus saving money and time in later stages of development when
requirements issues would otherwise appear. This automated analysis occurs in
one of 2 ways: Analysis or Checking.

I feel the most significant contribution of the paper is Alloys syntax. Like
the author states, it is small and free from un-necessary syntactic bloat. This
makes it easy to learn and apply the language in a relatively short period of time
(when compared to a much larger language such as Java for instance).

8) List 1-3 strengths of the paper. (1-2 sentences each,
identified as S1, S2, S3.)

S1: I liked how the author was upfront about some of the issues with Alloy; this makes
future work and enhancement of the language much easier for other researchers. And also
serves to alert those planning to use the language of the potential dangers.

S2: I feel like the syntax of Alloy is very clean, it nicely abstracts away the details
of the underlying solver used for analysis. Alloy reminds me of relational algebra
in databases, in that it nicely abstracts away the mathematical foundations in Set
Theory making it accessible to those without a strong background or mathematical inclination.

S3: I feel that the ASCII syntax is nice. I have no experience with Z, but the reliance
upon LateX would get annoying to me too (in the paper it states that many people felt this way).

9) List 1-3 weaknesses of the paper (1-2 sentences each,
identified as W1, W2, W3.)

W1: Something that I found interesting is the lack of frame conditions (page 279). The author
states that one possible solution is to introduce a keyword that can be used to
state that all unaltered state components remain unchanged from frame to frame.
The author gives good reasoning for why this is difficult, but I still feel like
there must be a way; what about just assuming that unaltered state components remain
unchanged?

W2: Another thing I was thinking about is the lack of string and integer support in Alloy. This
likely has something to do with the underlying SAT solver that is used, but I didn't find anywhere
that this is elaborated upon, it seems important but is thrown in as an afterthought.