#### Introduction

I could understand the problem being tackled, but showing concrete evidence of  
that problem with a motivating example would be important

the initial motivation was to merge feature models so that one can deal with  
large feature models in a modular way (by separately specifying smaller feature  
models and then using the proposed algorithm to merge them and obtain the  
desired feature model), but the evaluation seems to focus on merging different  
versions of similar features models.

#### Preliminaries

#### Semantics of Merging Operation

The semantics of the merging  
operation is defined by a relation between the configuration set of source and  
target FMs could not understand

First we represent a configuration of an FM as a set of  
features contained in the configuration. **Definition** **is** **circular**

The subset constraint at the beginning of Section 3.3 does not define the  
semantics of the merging operation, but some property it must satisfy. Many  
operations might satisfy that, so how do I know which one this paper is  
proposing? The algorithm defines it, but the semantics of the operation should  
define it too, at a higher level. This is mostly a terminology issue, but  
should at least be discussed. More important, Table 1 shows that the expected  
result in **Fig 2 would not strictly satisfy the subset constraint; that is, the  
target in Fig 2 is not a proper merge following the 'semantics' of the merging**  
operation. In fact, {Screen, High Resolution, Touch} is not a valid  
configuration of the target FM. **High Resolution is not even a feature of the  
target FM.**

#### Merging Algorithm

"The algorithm contains two manual steps and two automated steps"  
  
There is no such thing as manual steps in an algorithm (algorithms is a  
procedure for computation!). You probably talking about procedure, or a method,  
or a workflow, not an algorithm.

**Cloning parent incompatible features creates an incorrect model with respect to  
your semantics**

**I do think however that this issue can be toned down, by putting more  
emphasis on the entire workflow and not claiming that the algorithm solves this  
problem, but it enables the user to solve it.**

I like the idea of enriching the hierarchies with refinement  
information. This reminds me of the idea that feature models  
are "views on ontologies" [see reference below]. In a sense,  
the annotations reveal some of the underlying ontology,  
which helps making more informed merge decisions.  
<http://gsd.uwaterloo.ca/node/76>

In the algorithm section, as you did with the semantics of the merging  
operation, I would prefer to see an example first,

not being able to understand the rationale for the approach  
used to merge constraints

It is important to explicitly mention earlier that 'the root feature of both  
sources must be matched'.

Lines 11-13 in the algorithm do not seem to match what is illustrated in Fig 5,  
which shows how to merge children of different semantic dependencies. Is the  
name get\_children\_of\_the\_same\_refinement\_semantics really appropriate?

In page 7, I understand that Another property is that the rules ensure the configuration set of target FM  
is **exactly** the cross-product of source FMs' configuration sets, not hold for 5 (b) because new **intermediate** **features** were introduced and they  
appear in the target configurations. The **formal** **definitions** should then be  
changed.

The example given in section 4.6 is not clear. Isn't it the general rule for  
cross-product semantics? I failed to notice anything particular about this  
example, regarding constraints.'

1) Features are matched purely on name. The authors don't address matching of  
 features themselves (e.g. synonyms or very similar features).  
2) It handles only requires and excludes cross-tree constraints - does not  
 address other forms of constraints.  
  
Both points could be handled same way as in the related work on reverse  
engineering  
feature models (<http://gsd.uwaterloo.ca/node/322>).

#### Case Study

The evaluation section needs a **threats to validity** subsection.

I would also like to see the effort in time for  
performing the merging operations. This would be good to show if the manual  
steps aren't a threat for the algorithm.'

It would be good to make the feature model artifacts available online.'

It was not entirely clear to me how much one gains  
by these annotations, though. Thus, I would suggest  
a better discussion of the benefits and tradeoffs, as  
the annotations represent manual effort. In particular,  
I would like to see some example that compare  
the results of merging with and without the annotations  
and a discussion of what the annotations helped to do.

What were the cases that  
were handled well and what were the cases that caused problems.  
I would like to learn from the evaluation the potential limitations  
and ideas for improvement.

Also, the technique assumes  
manual preprocessing to add domain semantics to model prior  
merging, but does not evaluate the effort required to do this.

But it does not check how much  
information in the merge is missing (lost constraints, etc). How the merged  
models compare to models that would be created manually to represent the same  
problems? Also, the evaluation is not against the baseline of the existing  
algorithms, so we cannot really see whether the current algorithm is an  
improvement. What are the differences with other merging algorithms for FMs ?

#### Typos

pg 10: (FS: Feature Set) better to explain in the text, not in the table

- Figures should appear at the top or bottom of a page.  
- Algorithm 1. What happens when merge returns null?

- Section 1, page 2. "acceptable features" - what does this mean?

Section 4.3 is  
essentially incomprehensible, even though it appears trivial after 2-3  
readings.  
  
  
The definiton of the FS function is extremely well hidden in Table 3. It should  
be introduced in the text (either in section 2 or in section 3).

The introduction part is not clear enough, e.g. the concepts of  
cross-product semantics and rich-refinement types are mentioned without giving  
the definitions first.

   - Not enough information on the evaluation part. It would be nice to  
refer to a repository where all the source feature models and target models in  
the experiments are made available.