EMWDA Working Group on Lossy Transformations

Discussion Report

Editor: Anneke Kleppe

People Attending

Ed Willink, Jim Steel, Joao Paulo Almeida, Octavian Patrascoiu, Dave Akhurst, Julian Johnson, Audris Kalnins, Olivier Le Merdy, Kevin Dockerill, and Anneke Kleppe.

Lossy Transformations

We started of with a number of different interpretations of the term 'Lossy transformation'. 'Lossy' could mean either of:

1. Necessary information is not present in input model (such as requirements or design intent that were never modelled)
2. Potential corruption by a transformation that does not satisfy its specification, or whose implementation is flawed (*invalid* transformation).
3. Deliberate discard of information in the input model, for instance because it cannot be represented in the output language (*partial* transformation).
4. Lack of traceability, i.e. the elements of the output model cannot be linked to the elements in the input model they were generated from.
5. Lack of reversibility, i.e. the input model cannot be restored from the output model.
6. Loss of info on how or why the transformation is executed.

To get a better grip we propose the use of the term *invalid* (versus *valid*) for the second meaning, and the term *partial* (versus *complete*) for the third meaning. A transformation T may be invalid because either T does not apply to all possible input models or because there is a gap between the specification of T and implementation of T.

With regard to option one, incomplete input, we considered this to be the responsibility of the transformation it self. Either it should issue a warning before executing or it should not execute at all. On the topic of option four, traceability, we concluded that traceability is not a theoretically difficult issue. It is feasible, although in practice one may need very large machines to run the transformations on. In the discussion on option five, reversibility, it was amazing to see that none of us found this to be a very big issue. There were no dissenters from the perception that reversibility is only relevant for approximately 10% of the transformations. In the discussion on option six, loss of info on how or why the transformation is executed, there remained on open question: if a compound transformation fails (is *invalid*), how can you find the element that causes the failure?

Transformation Use Cases and Semantics Preserving Transformations

Tracy Gardener, in her keynote, presented a list of possible use cases for transformations. We discussed two of them in more depth in order to see whether they would need different types of transformations. We discussed pattern expansion and PIM to PSM transformations. Our conclusion was that the differences were not very large. For pattern expansion the transformation can be called 'in-place', which means that the source and target model are the same in some meaning, at least they are written in the same language. at this point in the discussion it became clear that we need to define equivalence of systems (and after that also of models) before we are able to define what a semantics preserving transformation is. Another conclusion was that parameterisation of transformations should be possible for any type of transformation.