

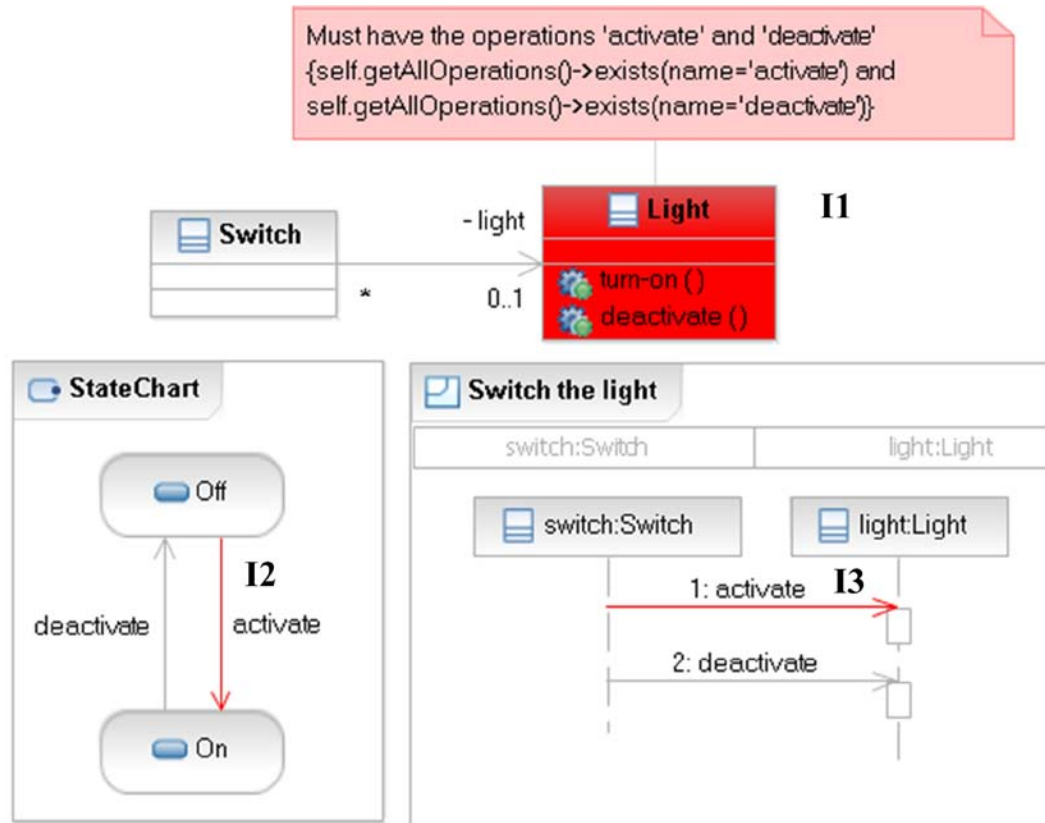


Utilizing the Relationships Between Inconsistencies for more Effective Inconsistency Resolution

Alexander Nöhrer
and
Alexander Egyed

Johannes Kepler University, Linz, Austria

Simple UML Example



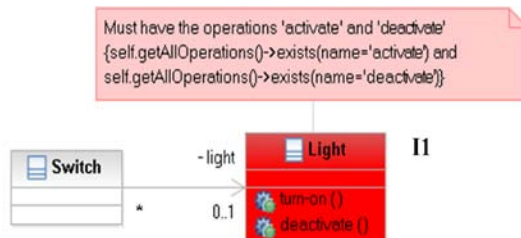
- Different constraints
- 3 inconsistencies (I1, I2, I3)
- In fact such a simple example a fix to all 3 inconsistencies is trivial

State-of-the-Art in Fixing

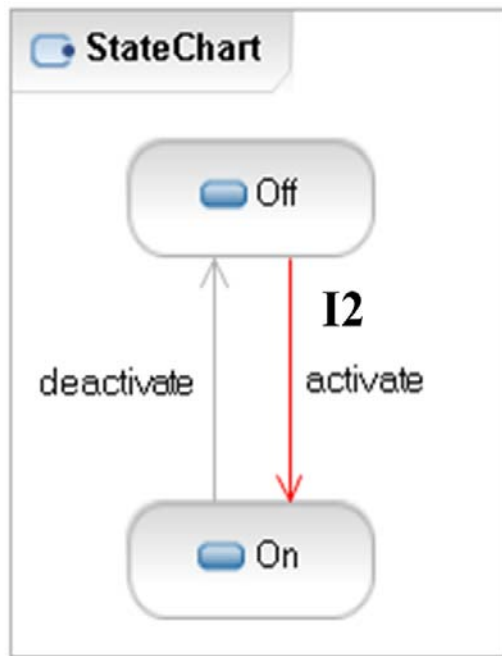
- Typically Inconsistencies are looked at one at a time
- Without considering relations between inconsistencies

Fixes for I1

- adding the operation *activate* to the class *Light* (F1)
- changing its operation *turn-on* to *activate* (F2)
- changing the model constraint C1 to fit the model (F3)

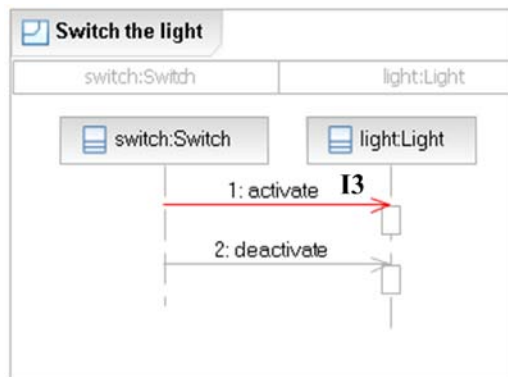


Fixes for I2



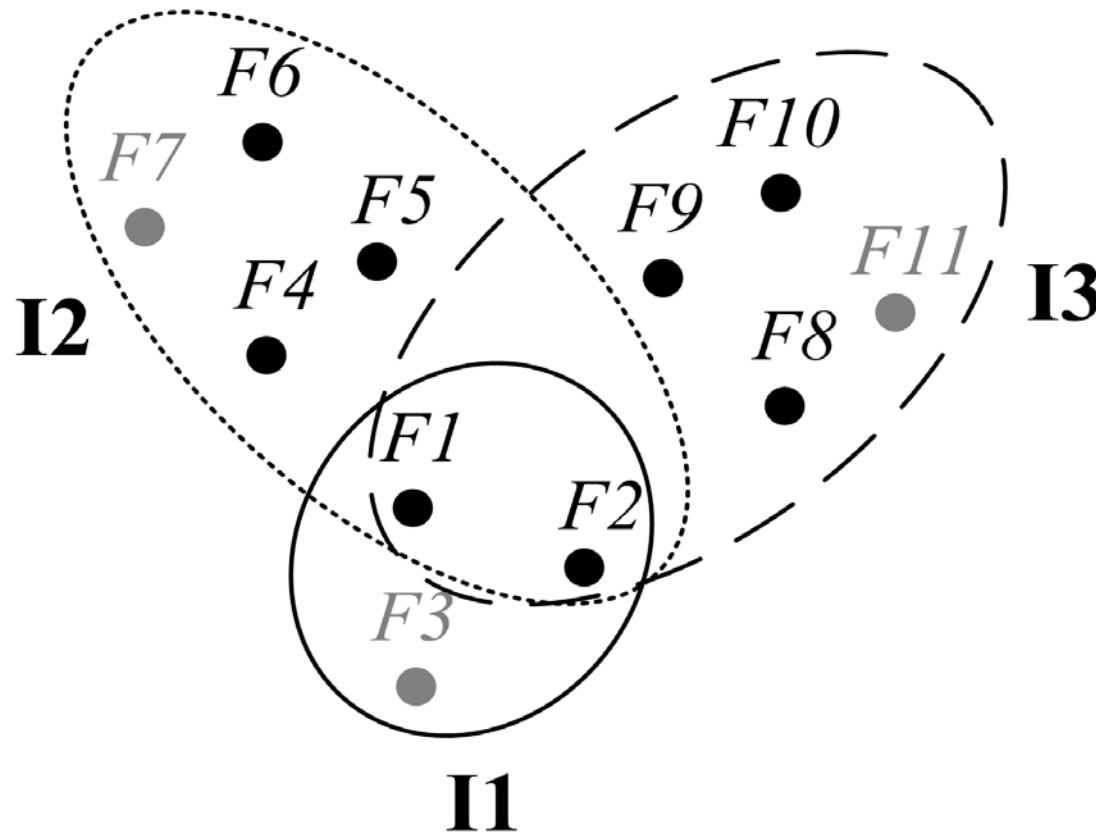
- adding the operation *activate* to the class *Light* (F1)
- changing its operation *turn-on* to *activate* (F2)
- change the inconsistent transition to *turn-on* (F4)
- or to *deactivate* (F5)
- or remove it (F6)
- the meta-model constraint C2 could be changed (F7)

Fixes for I3



- adding the operation *activate* to the class *Light* (F1)
- changing its operation *turn-on* to *activate* (F2)
- change the inconsistent message to *turn-on* (F8)
- or to *deactivate* (F9)
- or remove it (F10)
- change the meta-model constraint C3 (F11)

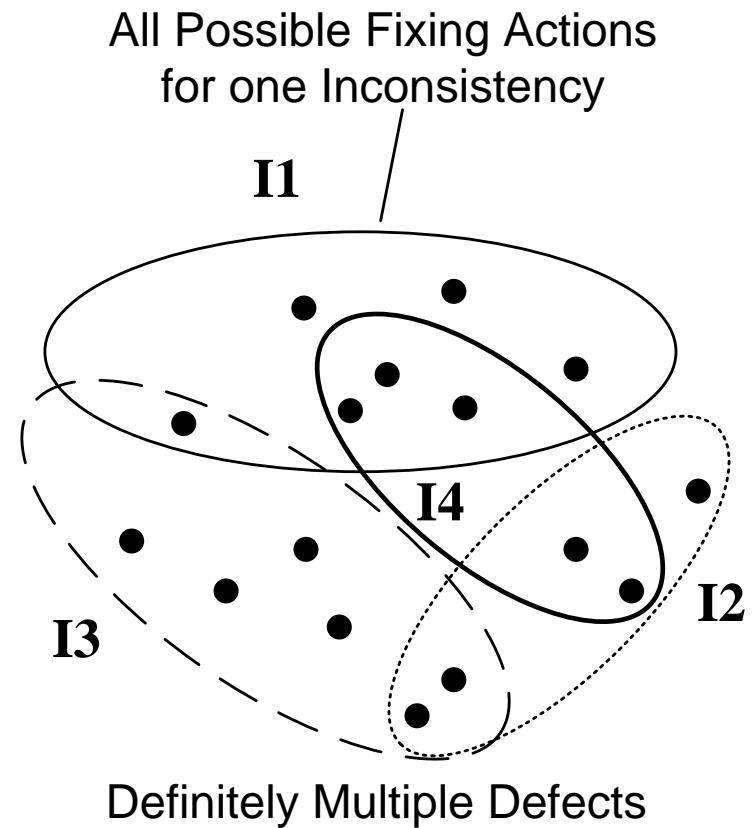
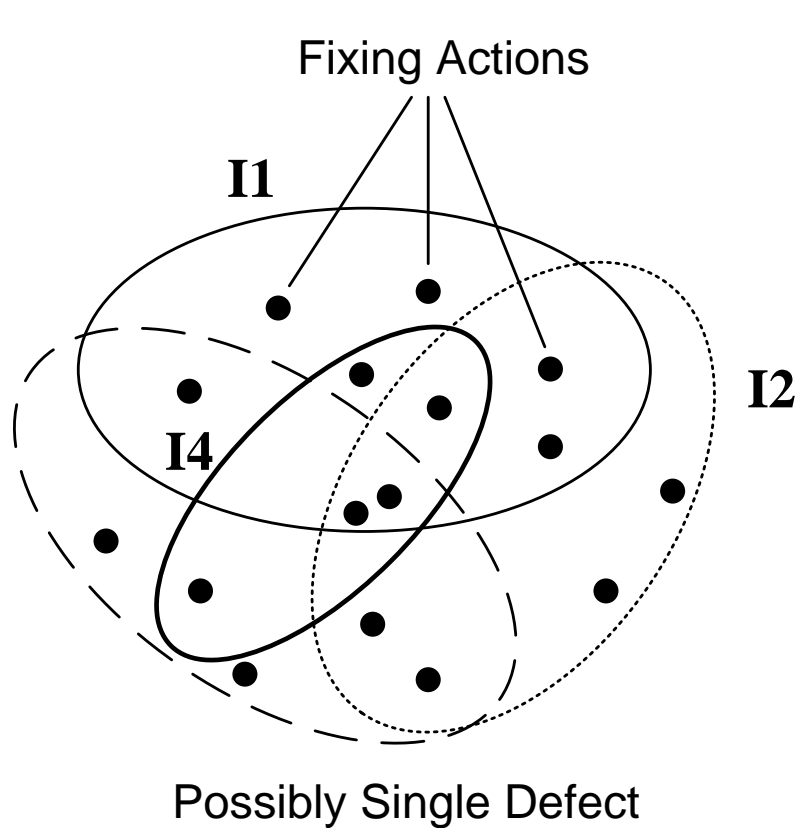
Interrelation of Fixes



Common Fixes:

- adding the operation *activate* to the class *Light* (F1)
- changing its operation *turn-on* to *activate* (F2)

Examples of different Interrelations



Vision

- Reduce the number of possible Fixes
- Increase scalability since the calculation of a fix can be cut-off as soon as a combined solution is not achievable any more
- More precise fixing actions since the impact of the changes onto a larger amount of model elements is already considered
- Supporting designers by unburden them of having to know about interrelations when choosing a fixing action



Proposed Approach (I)

- How often occur interrelated inconsistencies in real world examples?
- How many choices for fixing an inconsistency can be excluded considering these interrelations? How strong is this reduction?
- Can fixes become more concrete? If yes, how often does this happen?

Proposed Approach (II)

- If qualitative aspects proof to be useful
 - Automatically determine how inconsistencies are related with the help of model elements involved during the evaluation of violated constraints
- Concept of Trust
 - How can certain user interactions be interpreted and trusted respectively

Other Open Issues

- Are there different relationships between inconsistencies?
- Can more conceptual parallels be found to the compiler community and used?
- To what degree are AI concepts applicable (e.g. Clustering Algorithms)
- Impact on user guidance?

Summary

- Relationships between inconsistencies can and should be utilized to resolve inconsistencies!
- Still many questions to investigate

Questions



Alexander Nöhrer

Alexander Egyed

Johannes Kepler University
Institute for Systems
Engineering and Automation
Altenbergerstrasse 69
4040 Linz, Austria

alexander.noehrer@jku.at

alexander.egyed@jku.at