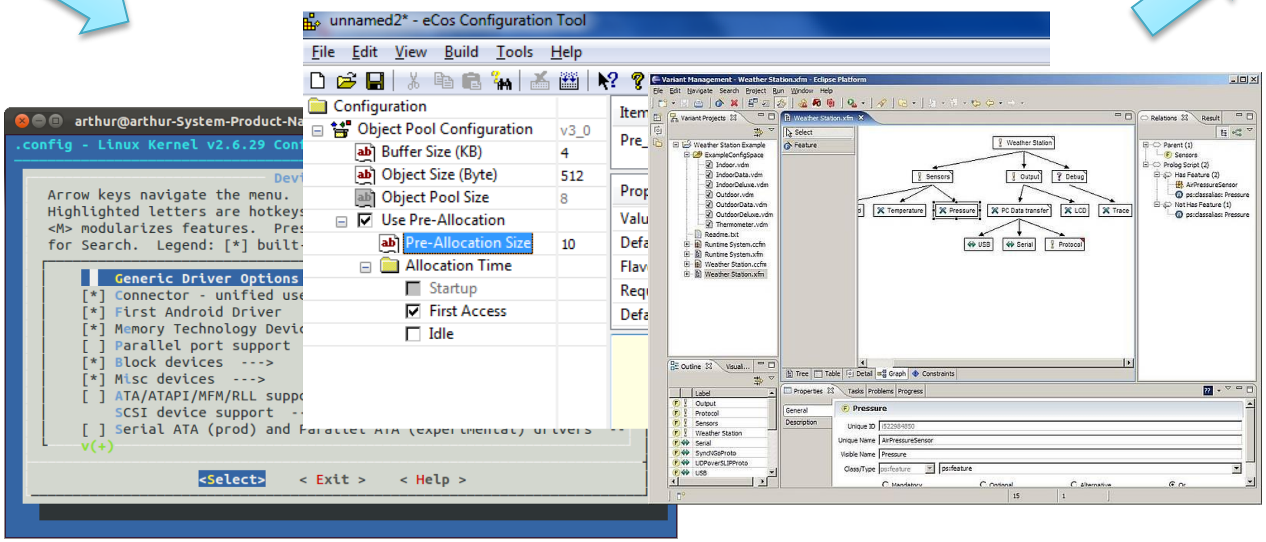
Generating Range Fixes for Software Configuration

Yingfei Xiong*	Arnaud Hubaux	Steven She Krzysztof Czarnecki
Peking University China	University of Namur Belgium	University of Waterloo Canada

* The work was done when Yingfei was at University of Waterloo

Variability Models & Configurators

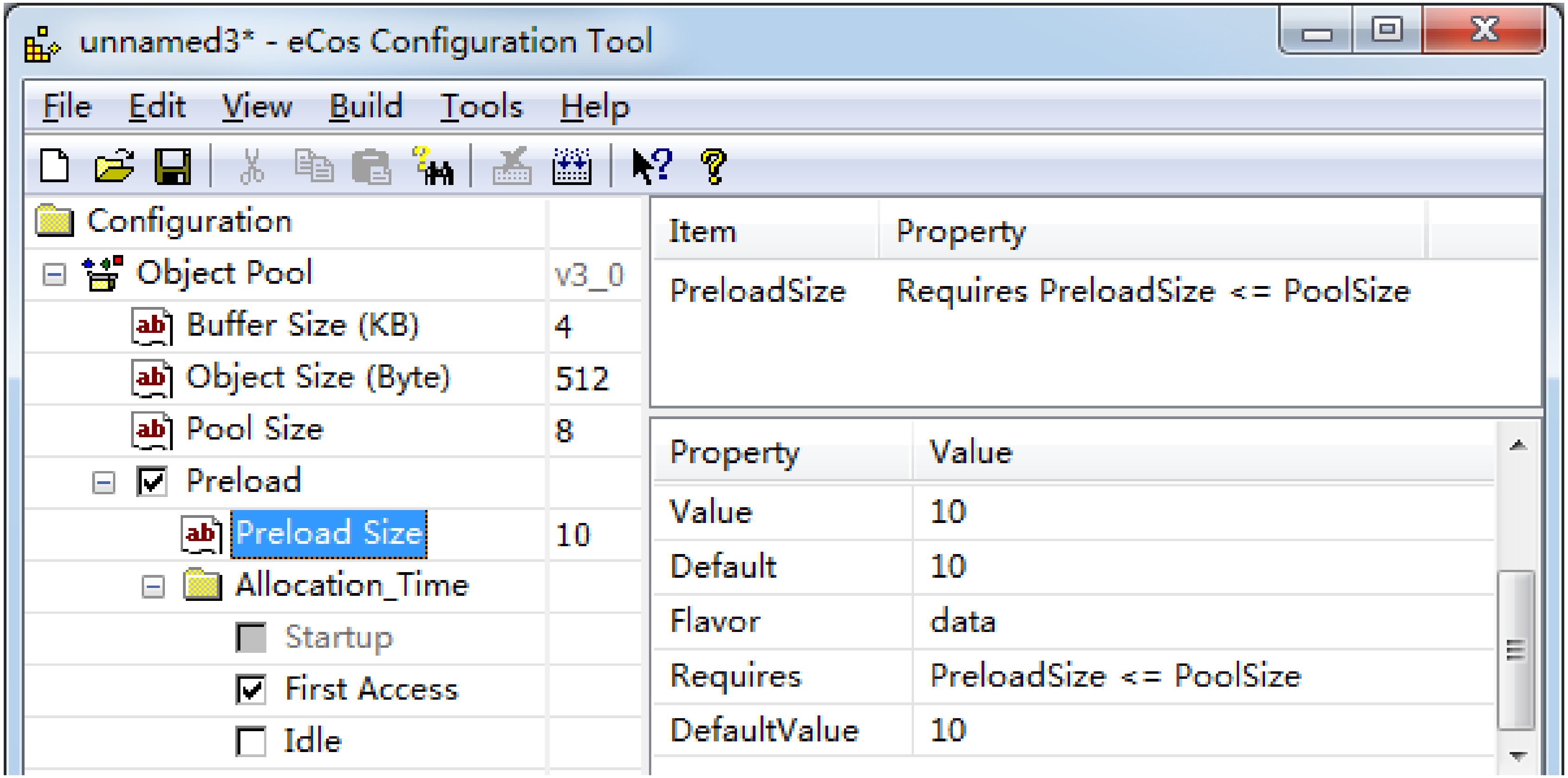
Configuration



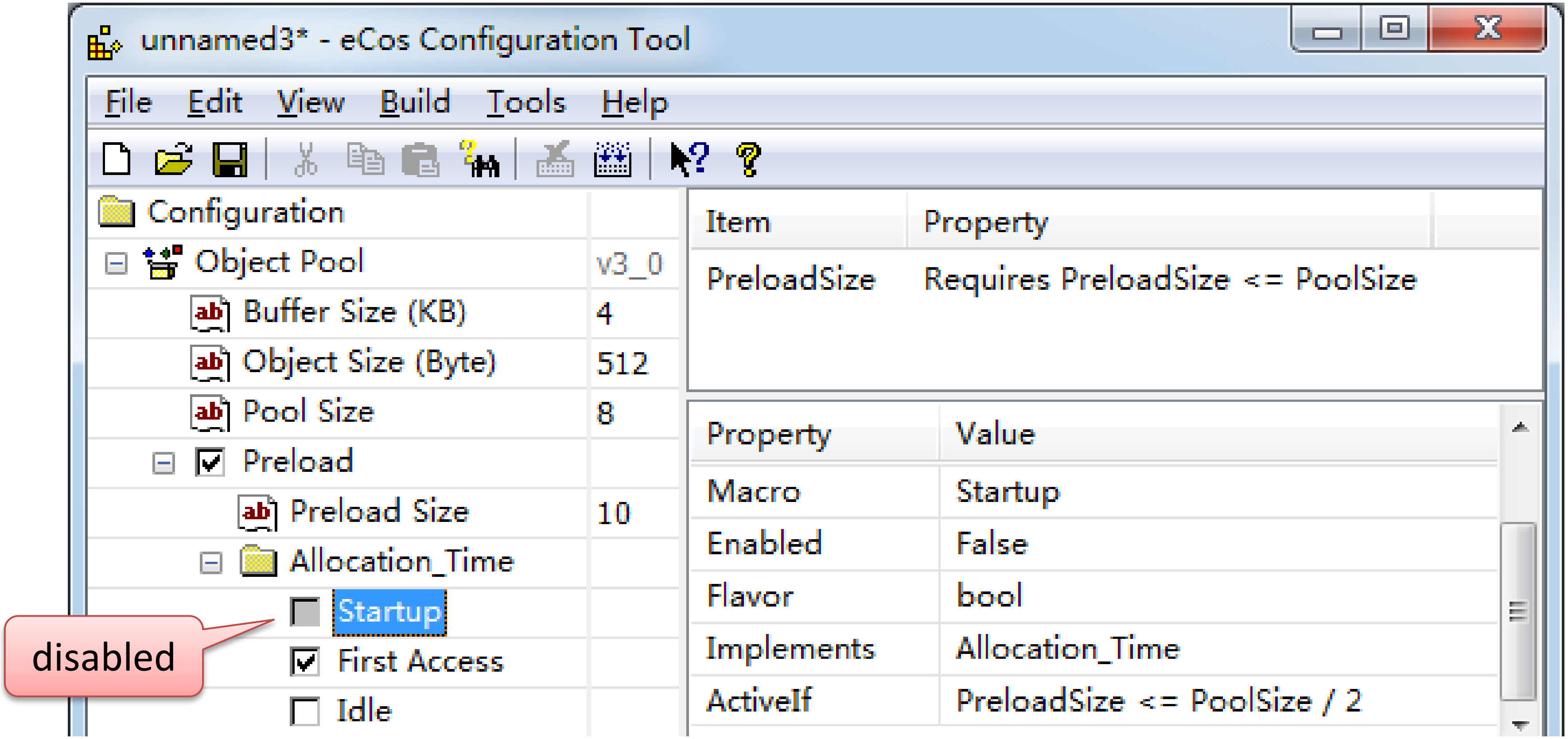
Linux Kconfig, eCos CDL, pure::variants, …

Variability Models

eCos Configurator - Errors



eCos Configurator - Inactive Options

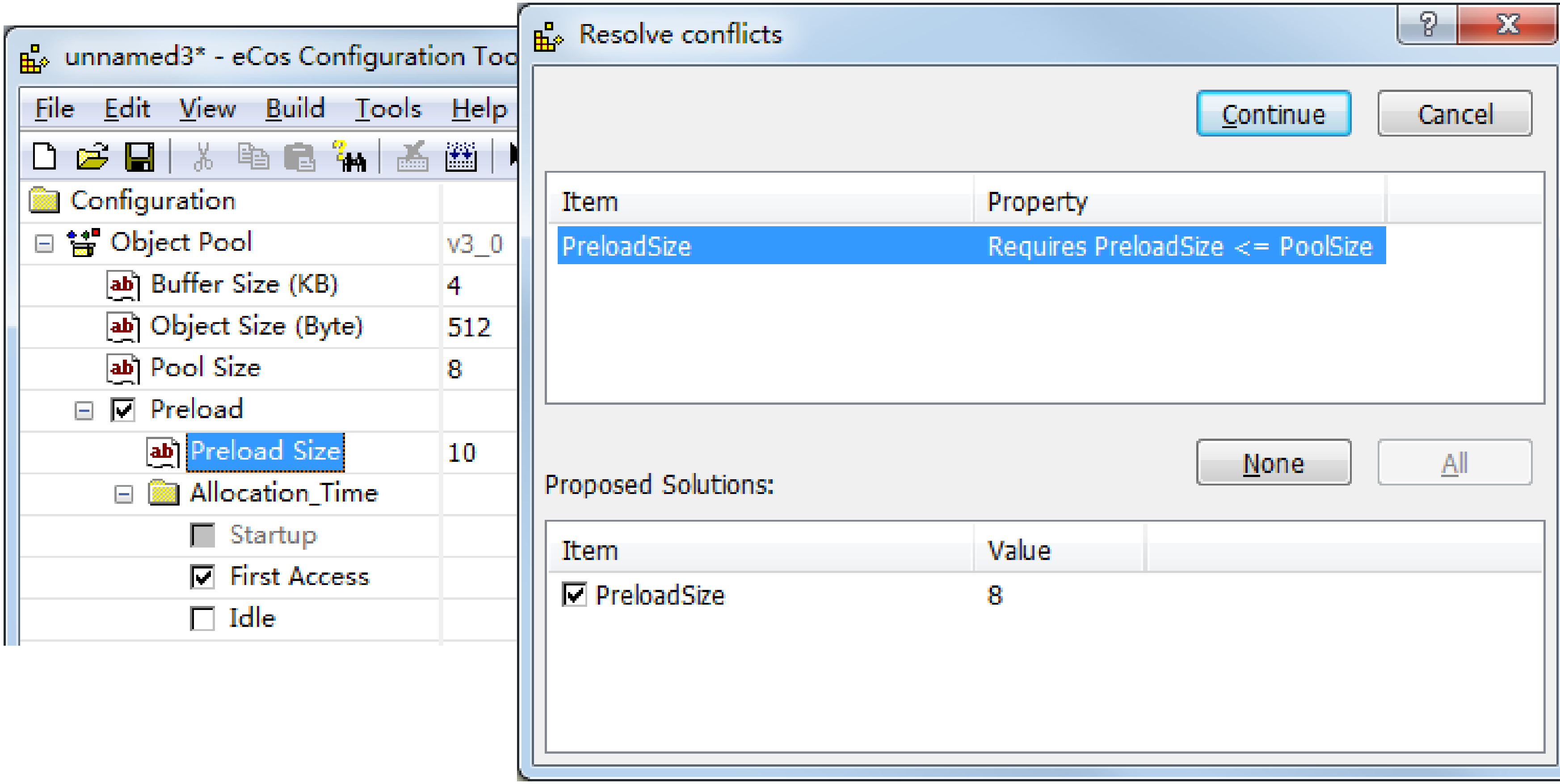


Error resolution and option activation both need to resolve violation of constraint.

Survey

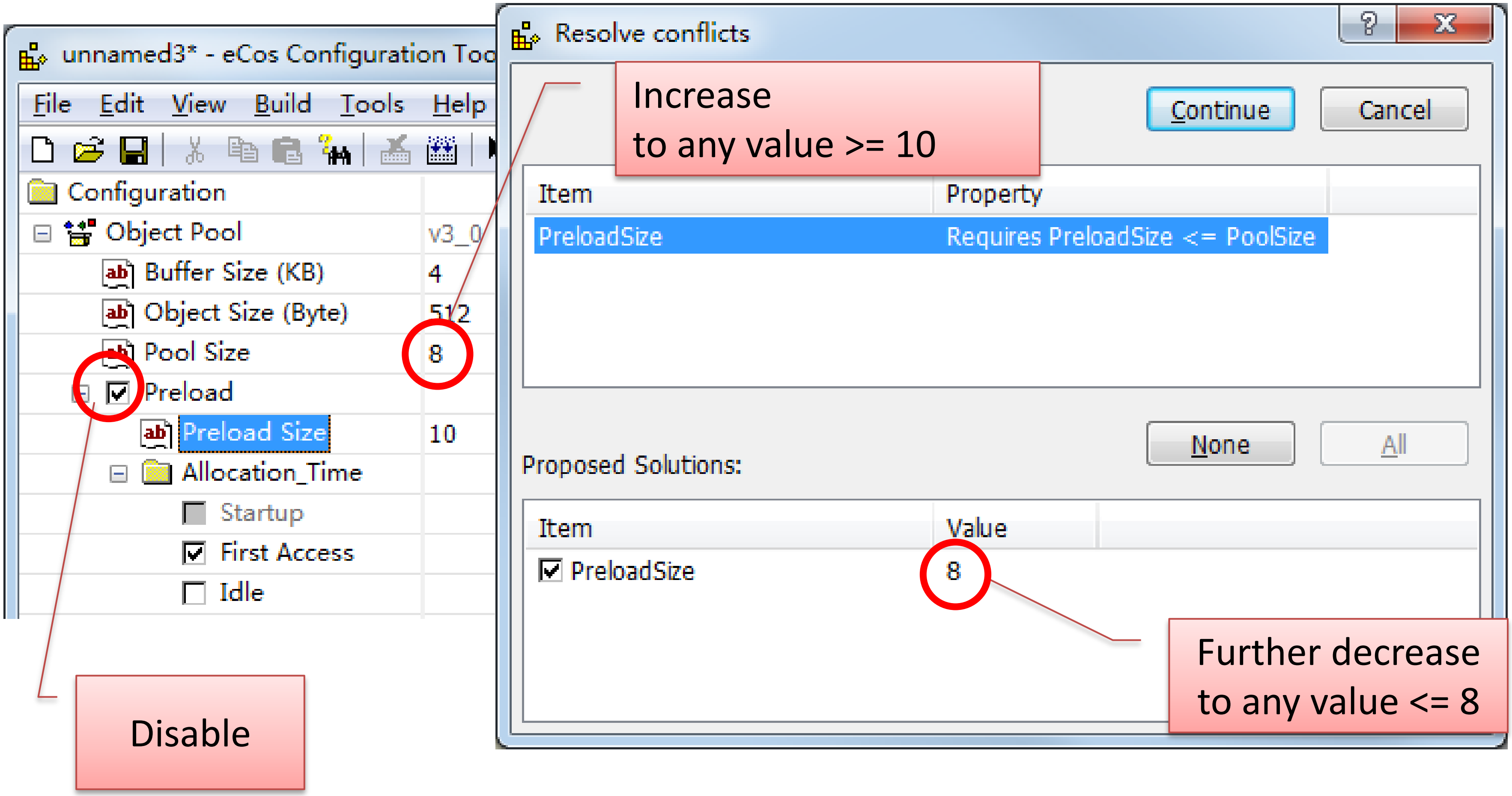
- 97 Linux users and 9 eCos users
- Resolving a violation is hard
  - 20% Linux users need "a few dozen minutes" to activate an option in average
  - 56% eCos users consider activation to be a problem

eCos Configurator



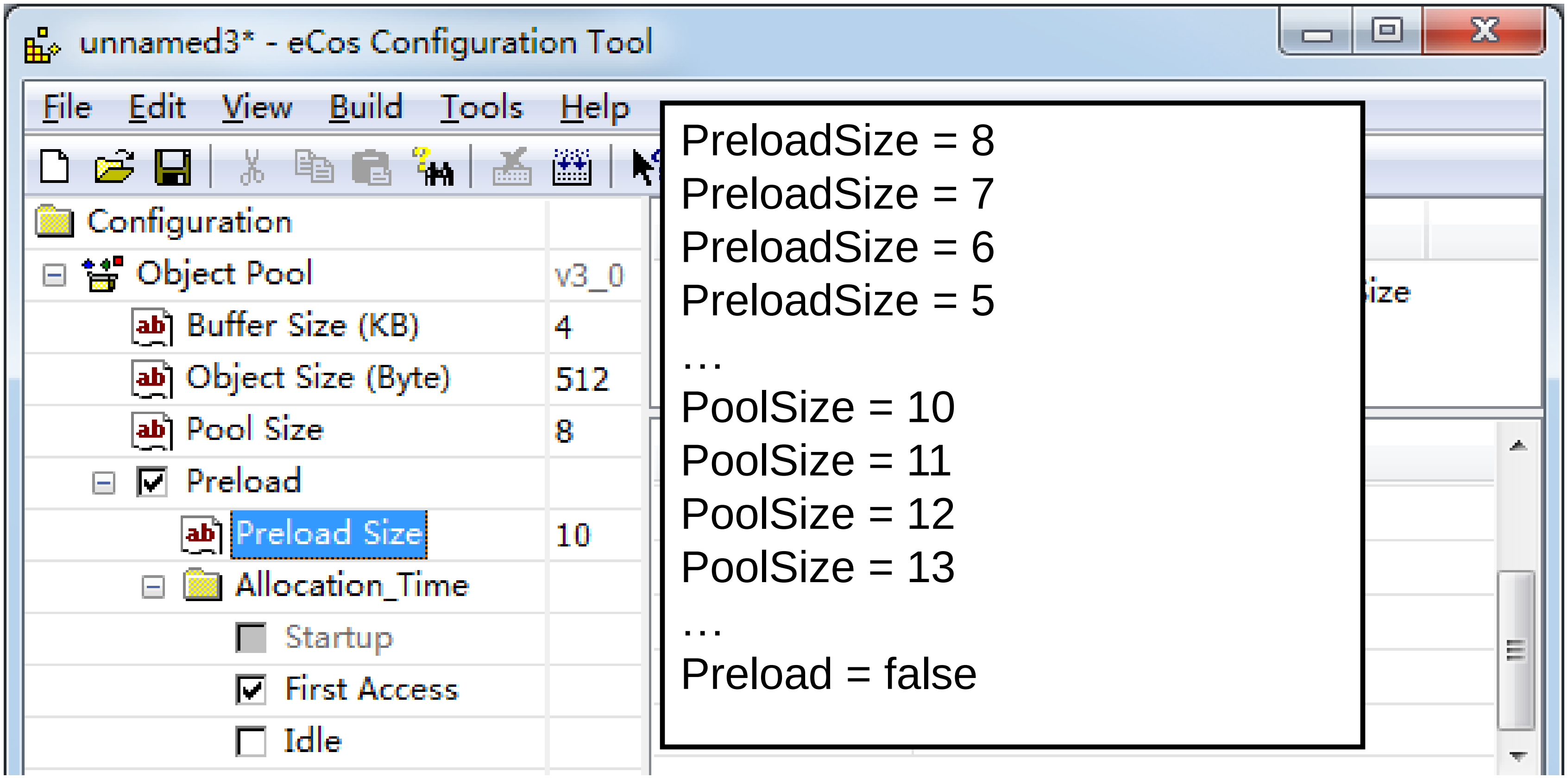
Essentially, fixes work for both resolving errors and activating options

Fix Incompleteness



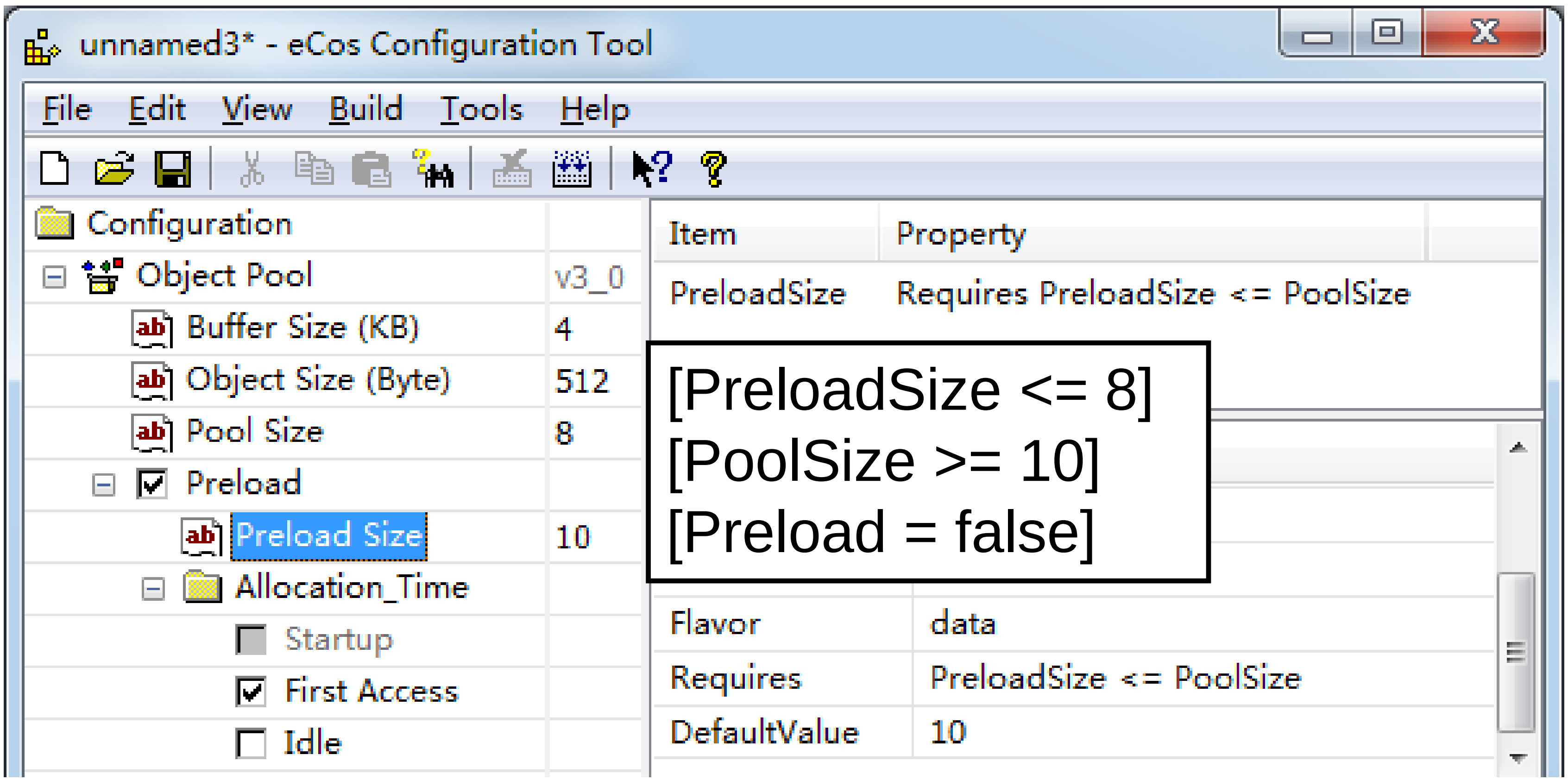
78% eCos users have ecountered situations where the proposed fix is not useful

How to complete fixes



PreloadSize = 8
PreloadSize = 7
PreloadSize = 6
PreloadSize = 5
…
PoolSize = 10
PoolSize = 11
PoolSize = 12
PoolSize = 13
…
Preload = false

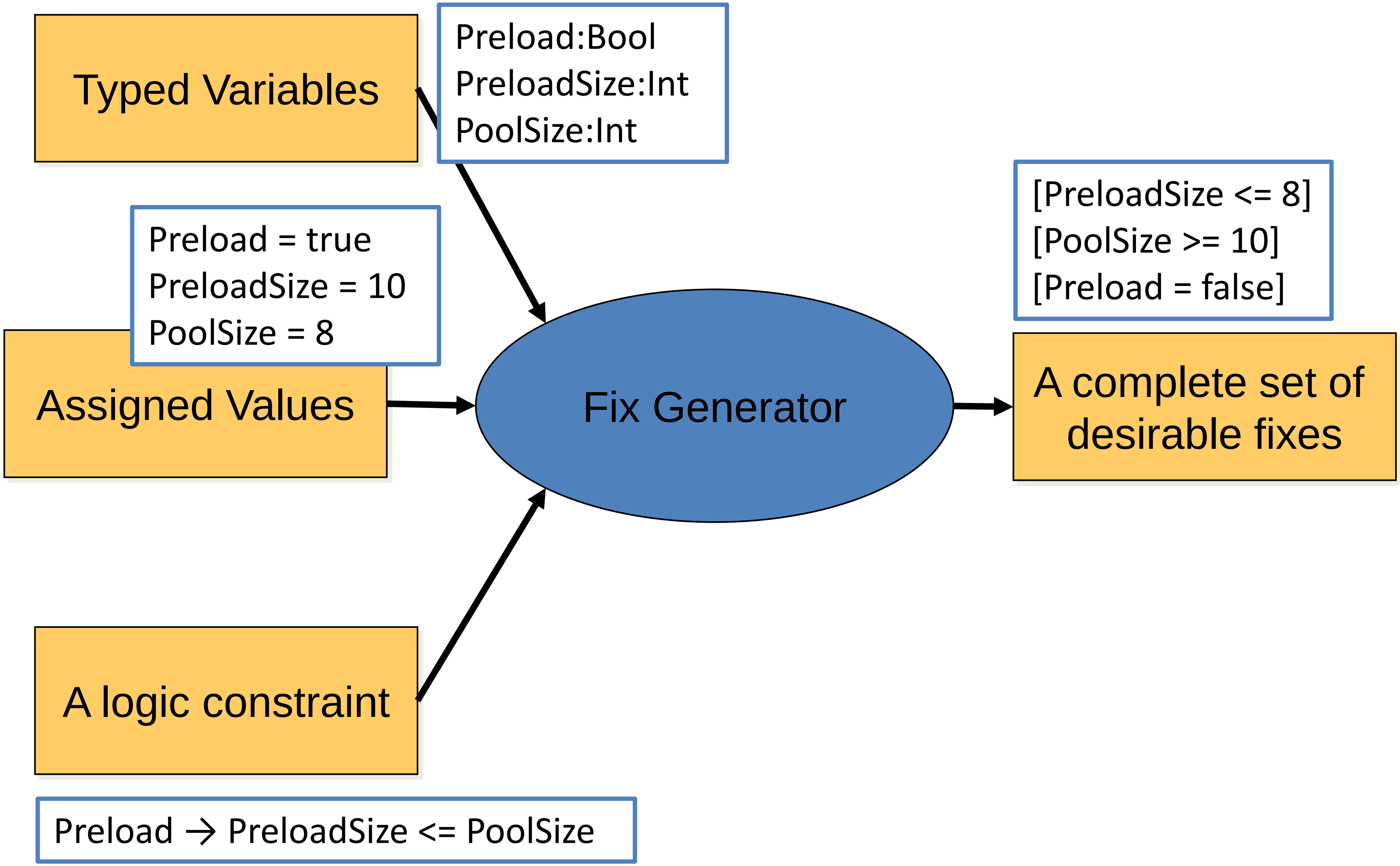
Our Solution – Range Fixes



Our Contributions

- Defining the range fix generation problem
  - Three desirable properties of range fixes
- Proposing a range fix generation algorithm
- Exploring the constraint interaction problem
  - Summarizing and adapting three strategies used in existing work
  - Comparing the strategies empirically

Fix Generation Problem – a General Definition



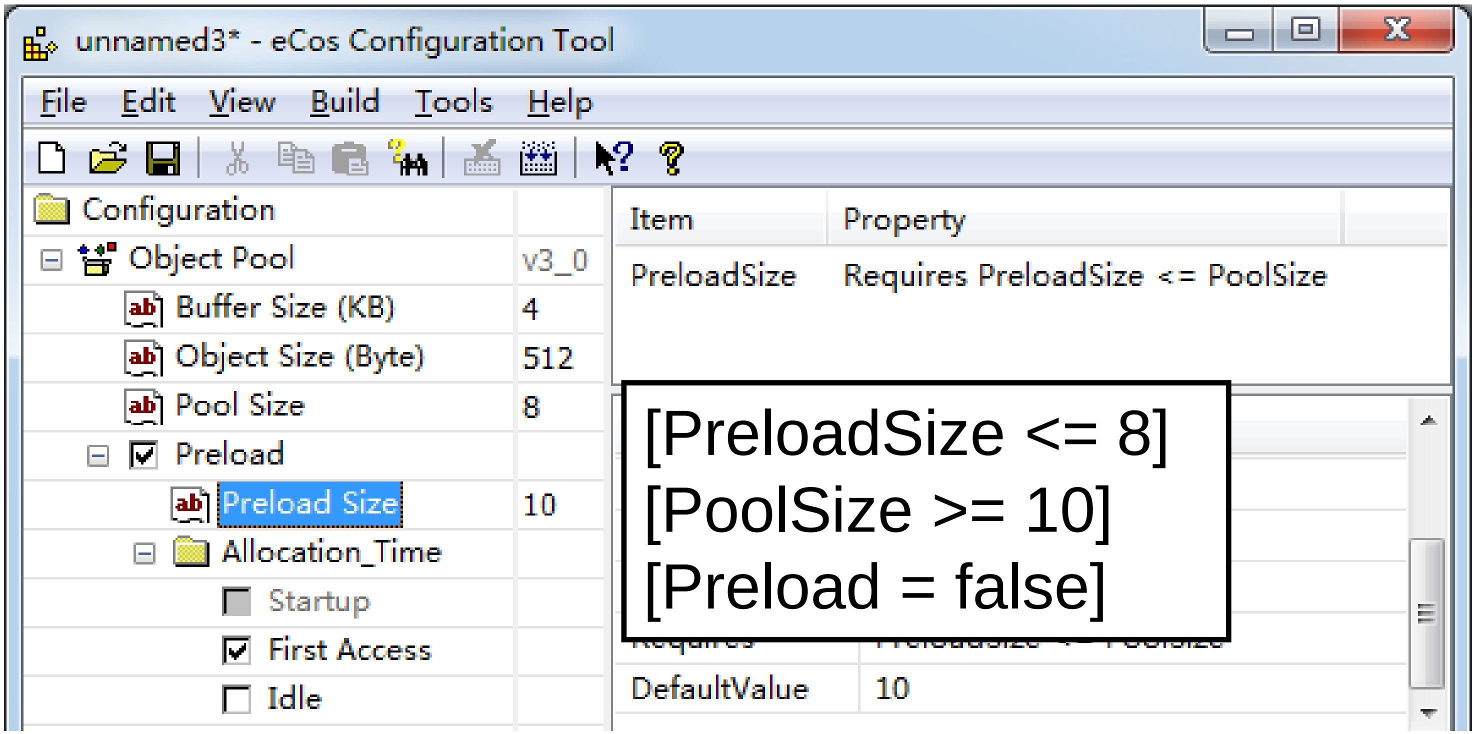
Desired Properties of Fixes

Correctness	Minimality of variables	Maximality of ranges
Any change represented by a range fix will satisfy the constraint	There is no way to change a subset of variables to satisfy the constraint	A range fix represents the maximal ranges over the variables
A desirable one: [PreloadSize <=8]		
Undesirable ones		
[PreloadSize <= 9]	[PreloadSize <=8, Preload = false]	[PreloadSize <=7]

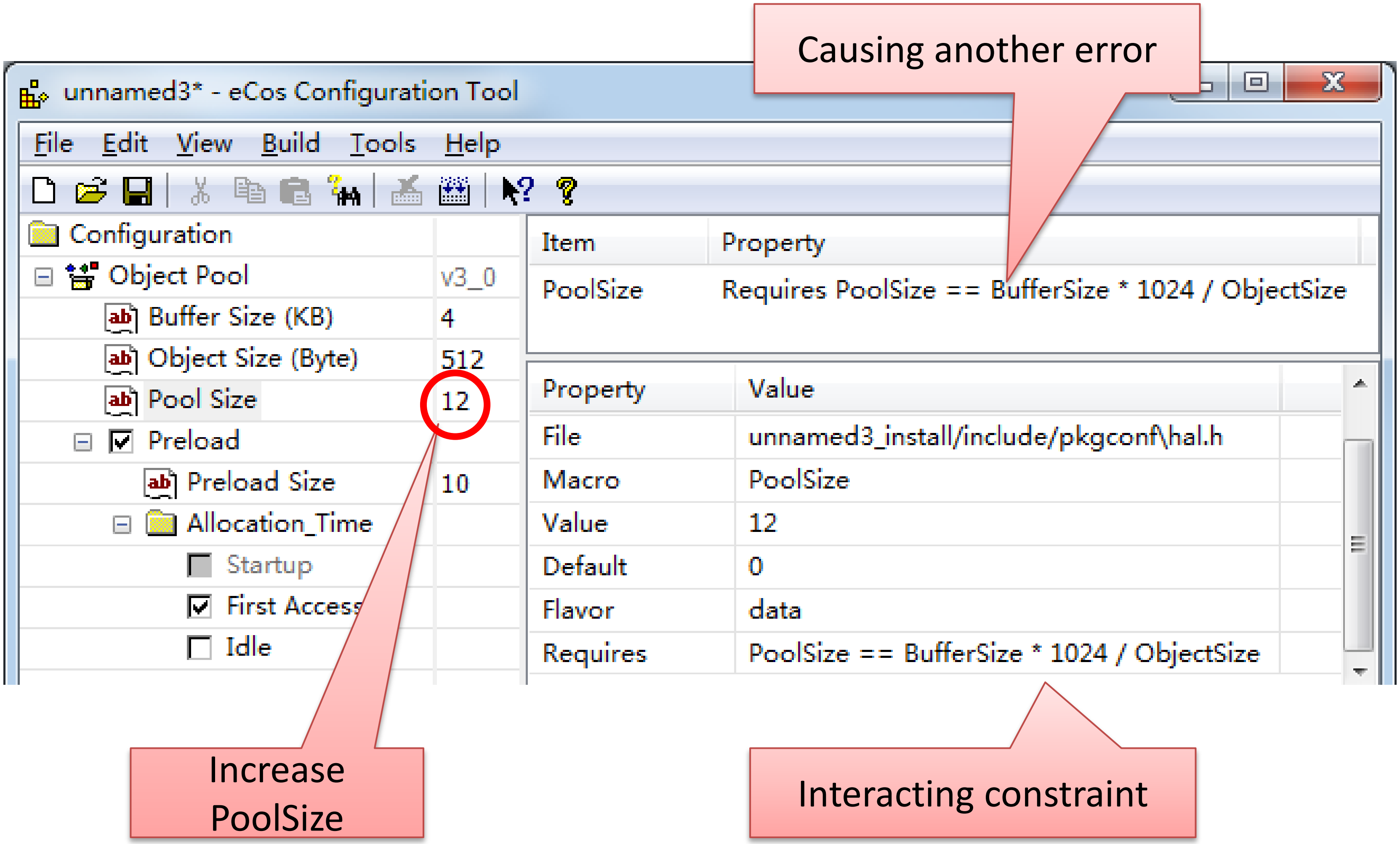
Algorithm

- Based on Reiter's theory of diagnosis
- Please check the paper for the details

Constraint Interaction

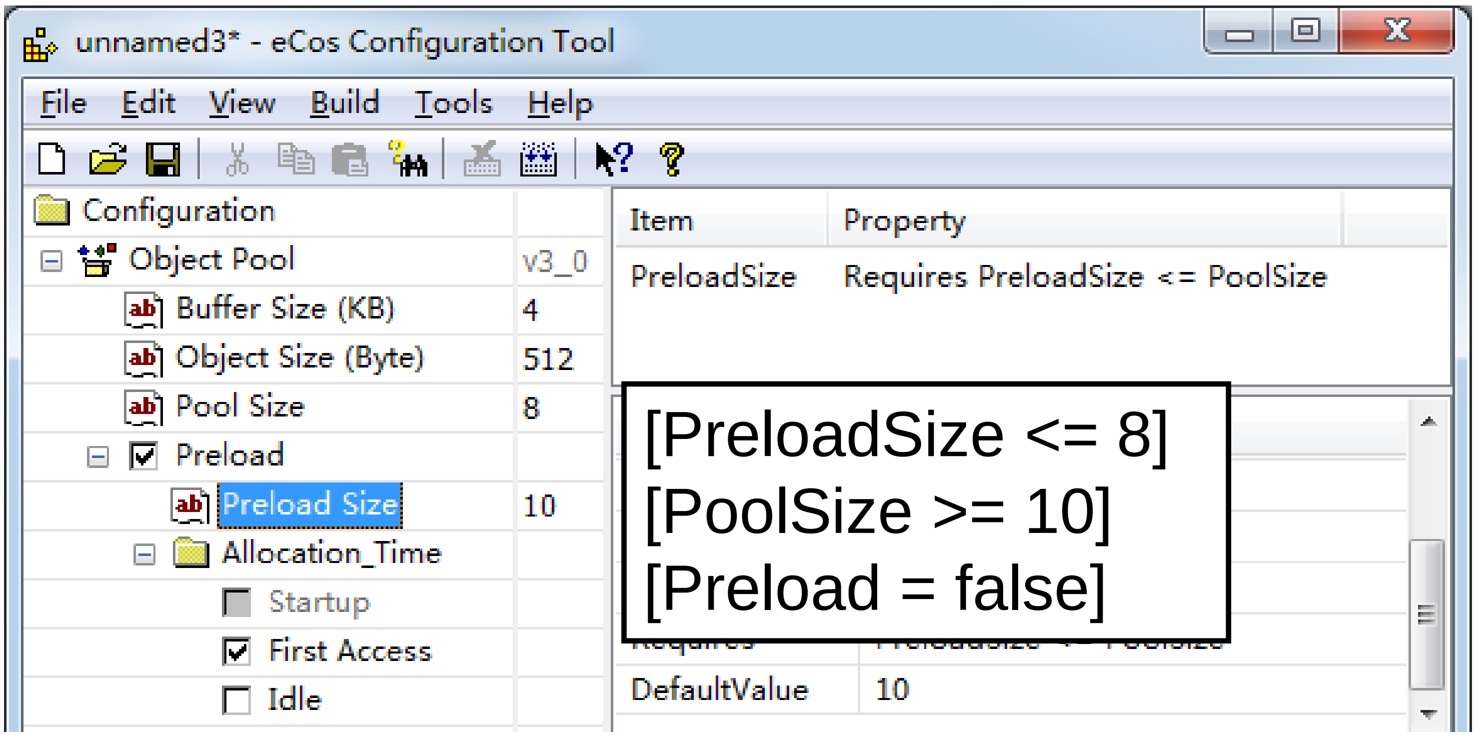


Constraint Interaction



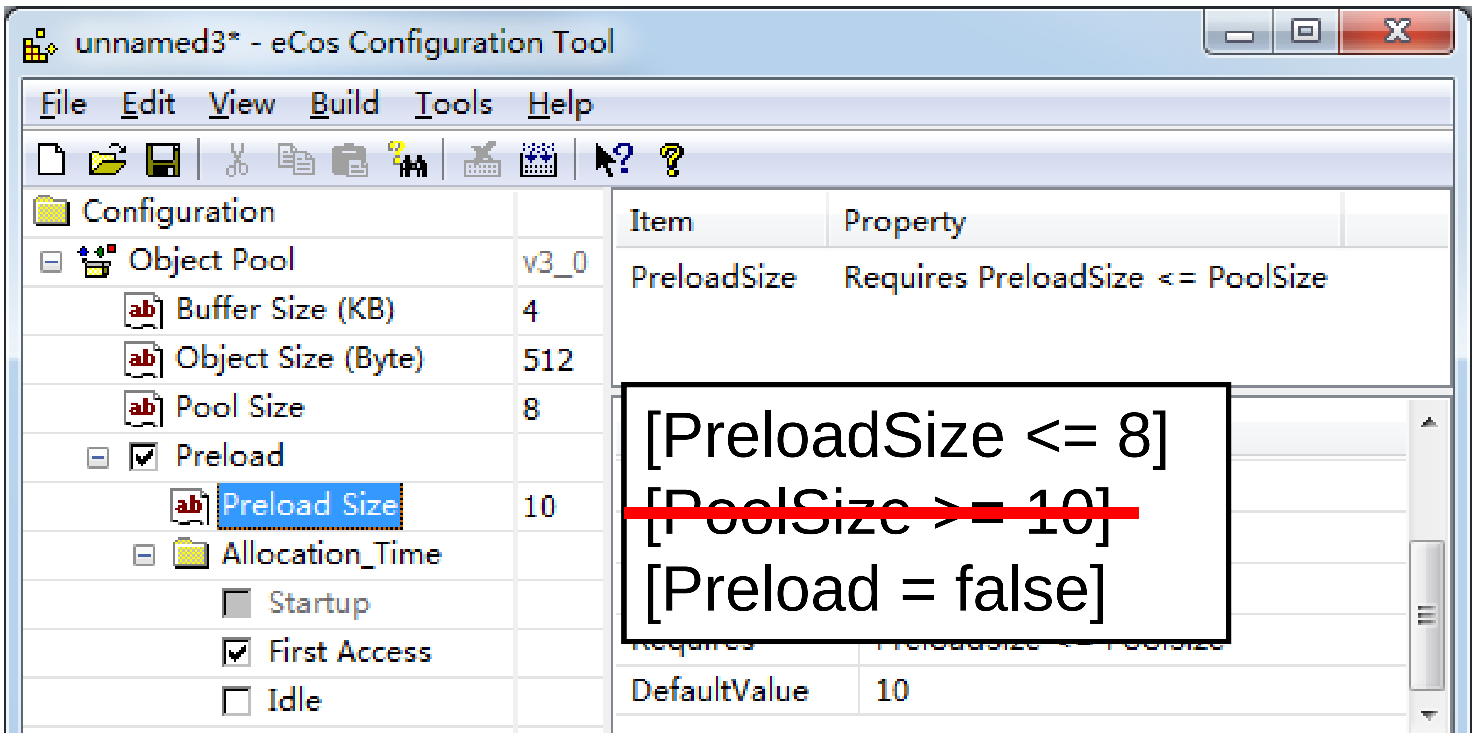
Ignorance

Ignore the interaction



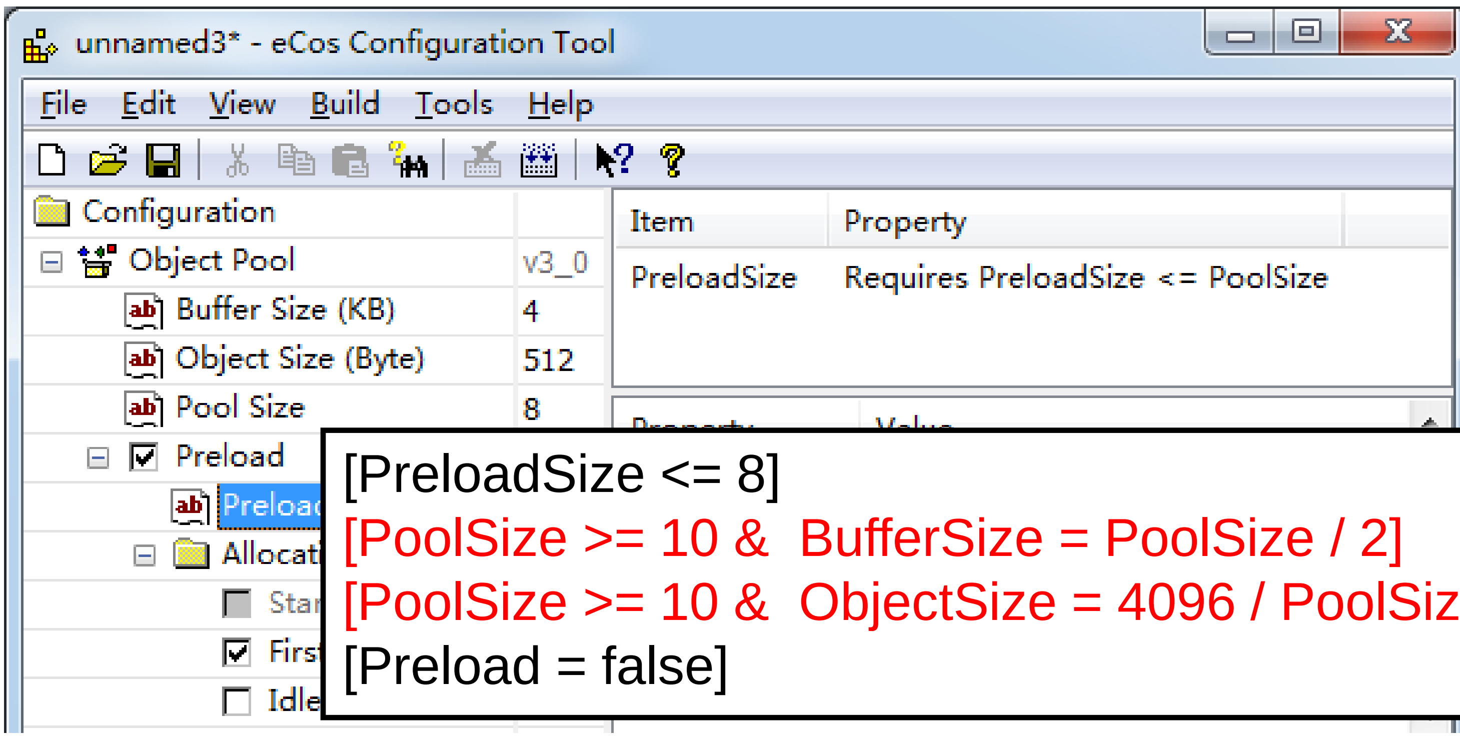
Elimination

Eliminate all changes that will violate other constraints



Propagation

Propagate the change along other constraints



[PreloadSize <= 8]
[PoolSize >= 10 & BufferSize = PoolSize / 2]
[PoolSize >= 10 & ObjectSize = 4096 / PoolSize]
[Preload = false]

Comparison of Strategies

	Ignorance	Elimination	Propagtion
Execution time	Shortest	Short	Possbily long
Complexity of fix lists	Simple	Simplest	Possibly complex
Introduction of new errors	Possible	Never	Never
Fix completeness	Complete (for one constraint)	Incomplete	Complete (for all constraints)

Experiments

- Source
  - Version histories from 5 open source projects
- Steps
  - Compare each pair of consecutive versions
  - Replay the user changes in different orders
  - Generate fixes for the violations and compare with user changes

Execution Time

	Ignorance	Elimination	Propagtion
Execution time	Average: 17ms Maximum: 20ms	Average: 20ms Maximum: 30ms	Average: 50ms Maximum: 250ms
Complexity of fix lists	Simple	Simplest	Possibly complex
Introduction of new errors	Possible	Never	Never
Fix completeness	Complete (for one constraint)	Incomplete	Complete (for all constraints)

Our algorithm is sufficiently fast for each strategy

Complexity of fix lists

	Ignorance	Elimination	Propagtion
Execution time	Average: 17ms Maximum: 20ms	Average: 20ms Maximum: 30ms	Average: 50ms Maximum: 250ms
Complexity of fix lists (Number of variables in a list)	Max: 4 Median: 2 Average: 2.2	Max: 4 Median: 2 Average: 1.64	Max: 58 Median: 2 Average: 8.0
Introduction of new errors	Possible	Never	Never
Fix completeness	Complete (for one constraint)	Incomplete	Complete (for all constraints)

In propagation, 83% of the fix lists contain less than 10 variables

Introduction of new errors

	Ignorance	Elimination	Propagtion
Execution time	Average: 17ms Maximum: 20ms	Average: 20ms Maximum: 30ms	Average: 50ms Maximum: 250ms
Complexity of fix lists (Number of variables in a list)	Max: 4 Median: 2 Average: 2.2	Max: 4 Median: 2 Average: 1.64	Max: 58 Median: 2 Average: 8.0
Introduction of new errors	44% of all violations	Never	Never
Fix completeness	Complete (for one constraint)	Incomplete	Complete (for all constraints)

Fix completeness

	Ignorance	Elimination	Propagtion
Execution time	Average: 17ms Maximum: 20ms	Average: 20ms Maximum: 30ms	Average: 50ms Maximum: 250ms
Complexity of fix lists (Number of variables in a list)	Max: 4 Median: 2 Average: 2.2	Max: 4 Median: 2 Average: 1.64	Max: 58 Median: 2 Average: 8.0
Introduction of new errors	44% of all violations	Never	Never
Fix completeness (coverage of user changes)	100%	57%	100%

eCos configurator: 73%

Conclusion

- Fix completenss can be achieved by organzing them into range fixes
- Range fixes can be generated automatically and efficiently
- Three strategies for constraint interaction
  - No absolutely best solution
  - Propagation strategy gives relatively better results than the other two

Thank you for your attention!

EccFixer: http://gsd.uwaterloo.ca/eccfixer