

Interactive fixes for software configuration

Yingfei Xiong, Peking University

Cooperation with:

Peking University

Bo Wang, Hansheng Zhang, Jie Wang,
Haiyan Zhao, Wei Zhang

University of Waterloo

Leonardo Passos, Steven She, Krzysztof Czarnecki

University of Namur

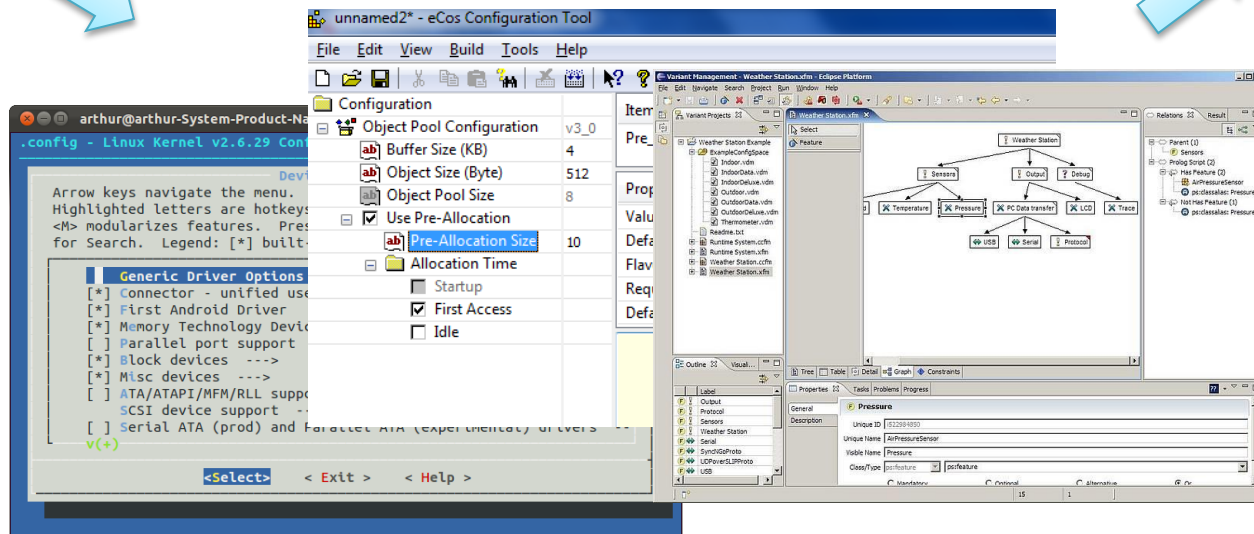
Arnaud Hubaux

2013

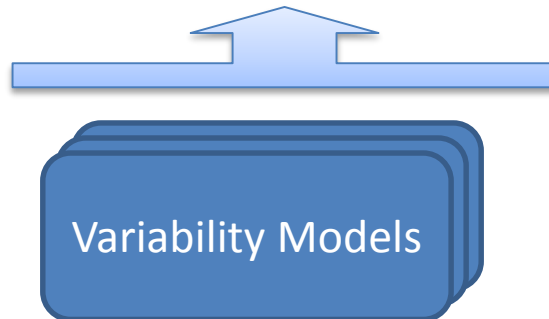
Variability Models & Configurators



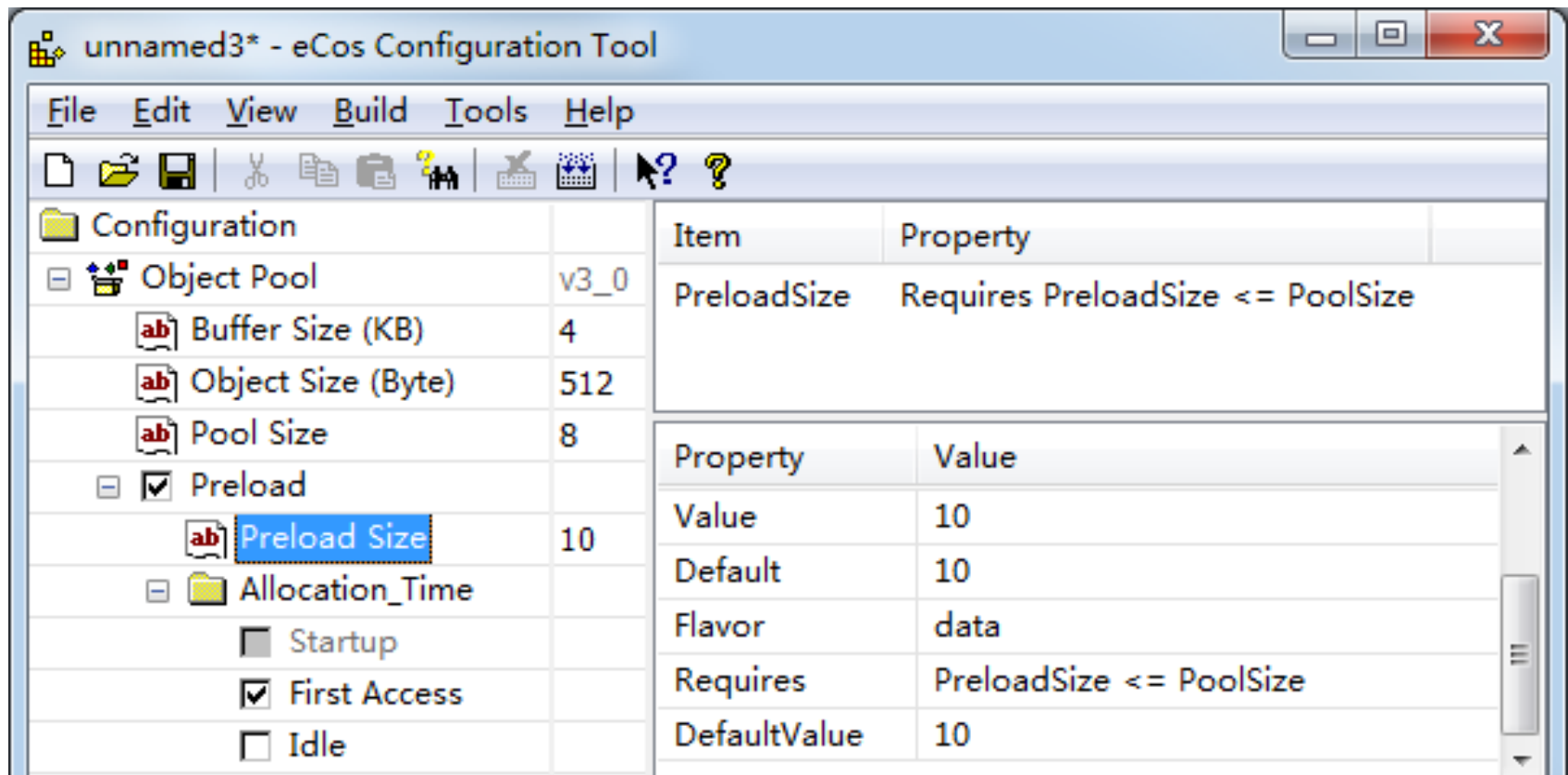
Configuration



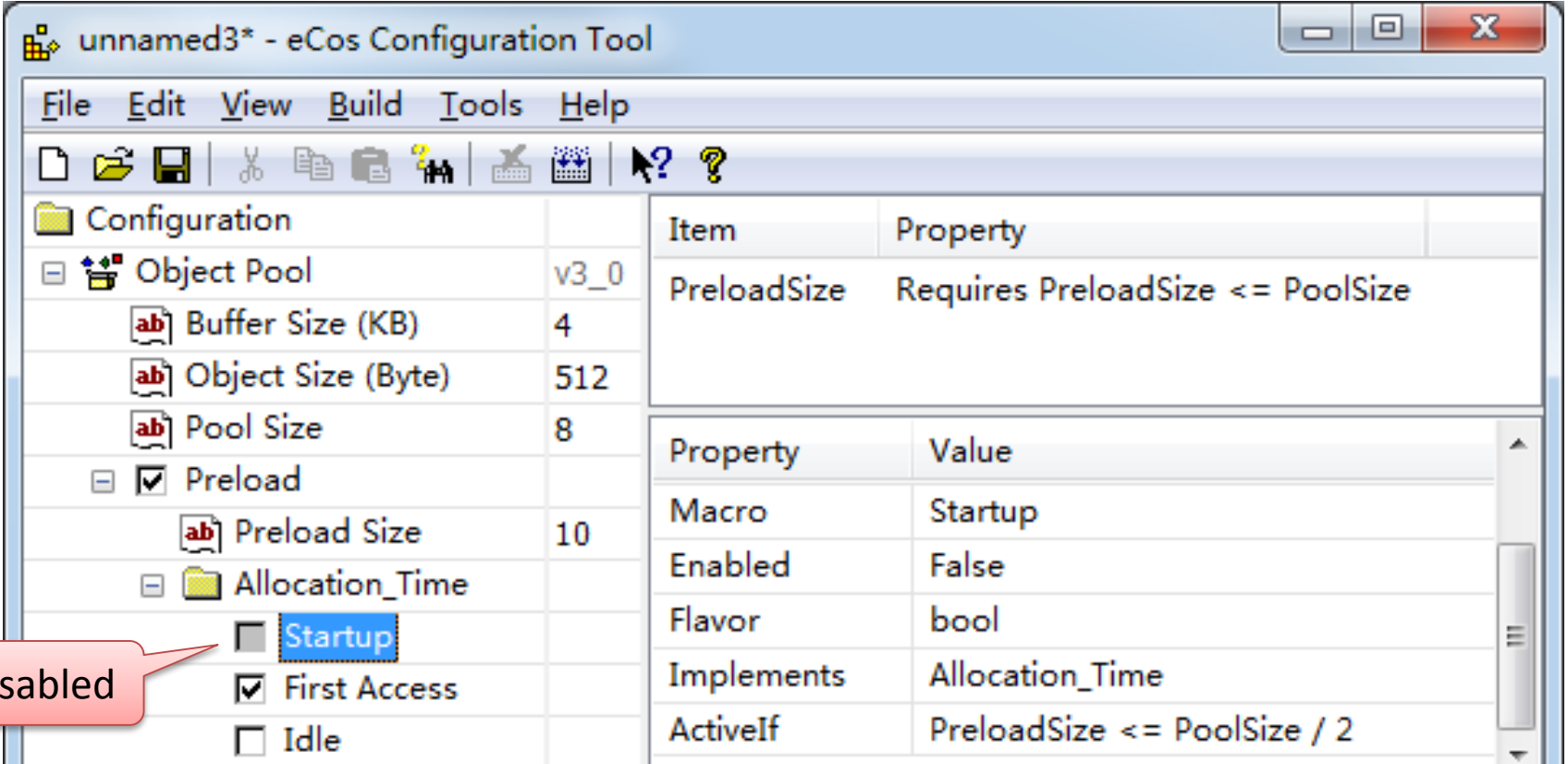
Linux Kconfig,
eCos CDL,
pure::variants,
...



eCos Configurator - Errors



eCos Configurator - Inactive Options



The screenshot shows the 'unnamed3* - eCos Configuration Tool' window. The left pane displays a tree view of configuration options. The right pane shows a table of properties and their values.

Configuration Tree:

- Configuration
 - Object Pool (v3_0)
 - Buffer Size (KB): 4
 - Object Size (Byte): 512
 - Pool Size: 8
 - Preload (checked)
 - Preload Size: 10
 - Allocation_Time
 - Startup (disabled)
 - First Access (checked)
 - Idle (unchecked)

Properties Table:

Item	Property
PreloadSize	Requires PreloadSize <= PoolSize

Property	Value
Macro	Startup
Enabled	False
Flavor	bool
Implements	Allocation_Time
ActiveIf	PreloadSize <= PoolSize / 2

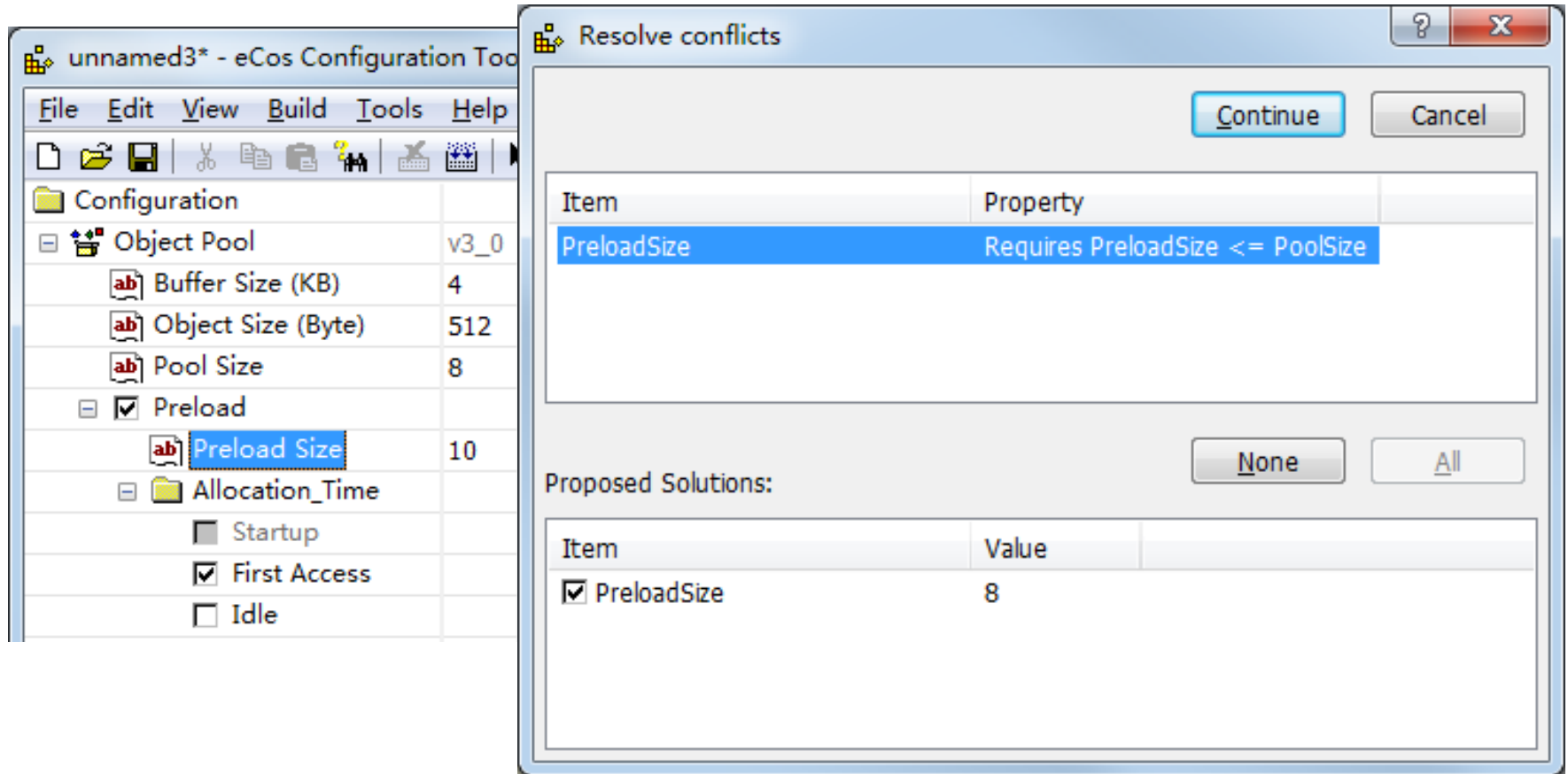
A red callout bubble with the text 'disabled' points to the 'Startup' option in the 'Allocation_Time' sub-tree.

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

The image shows a screenshot of the eCos Configuration Tool and a 'Resolve conflicts' dialog box. The configuration tool on the left shows a tree view with 'Object Pool' expanded, showing 'Buffer Size (KB)' (4), 'Object Size (Byte)' (512), 'Pool Size' (8), and 'Preload' (checked). The 'Preload Size' is set to 10. The 'Resolve conflicts' dialog on the right shows a conflict between 'PreloadSize' and 'PoolSize'. The dialog has a 'Continue' button and a 'Cancel' button. A red box highlights the 'PreloadSize' property with the text 'Increase to any value ≥ 10 '. The 'Proposed Solutions' section shows a table with 'Item' and 'Value' columns. The 'PreloadSize' item is checked with a value of 8. A red box highlights the value 8 with the text 'Further decrease to any value ≤ 8 '. A red box highlights the 'Preload' checkbox with the text 'Disable'.

unnamed3* - eCos Configuration Tool

File Edit View Build Tools Help

Configuration

- Object Pool (v3_0)
 - Buffer Size (KB) 4
 - Object Size (Byte) 512
 - Pool Size 8
 - ☒ Preload
 - Preload Size 10
 - Allocation_Time
 - ☐ Startup
 - ☒ First Access
 - ☐ Idle

Resolve conflicts

Continue Cancel

Item	Property
PreloadSize	Requires PreloadSize \leq PoolSize

Proposed Solutions:

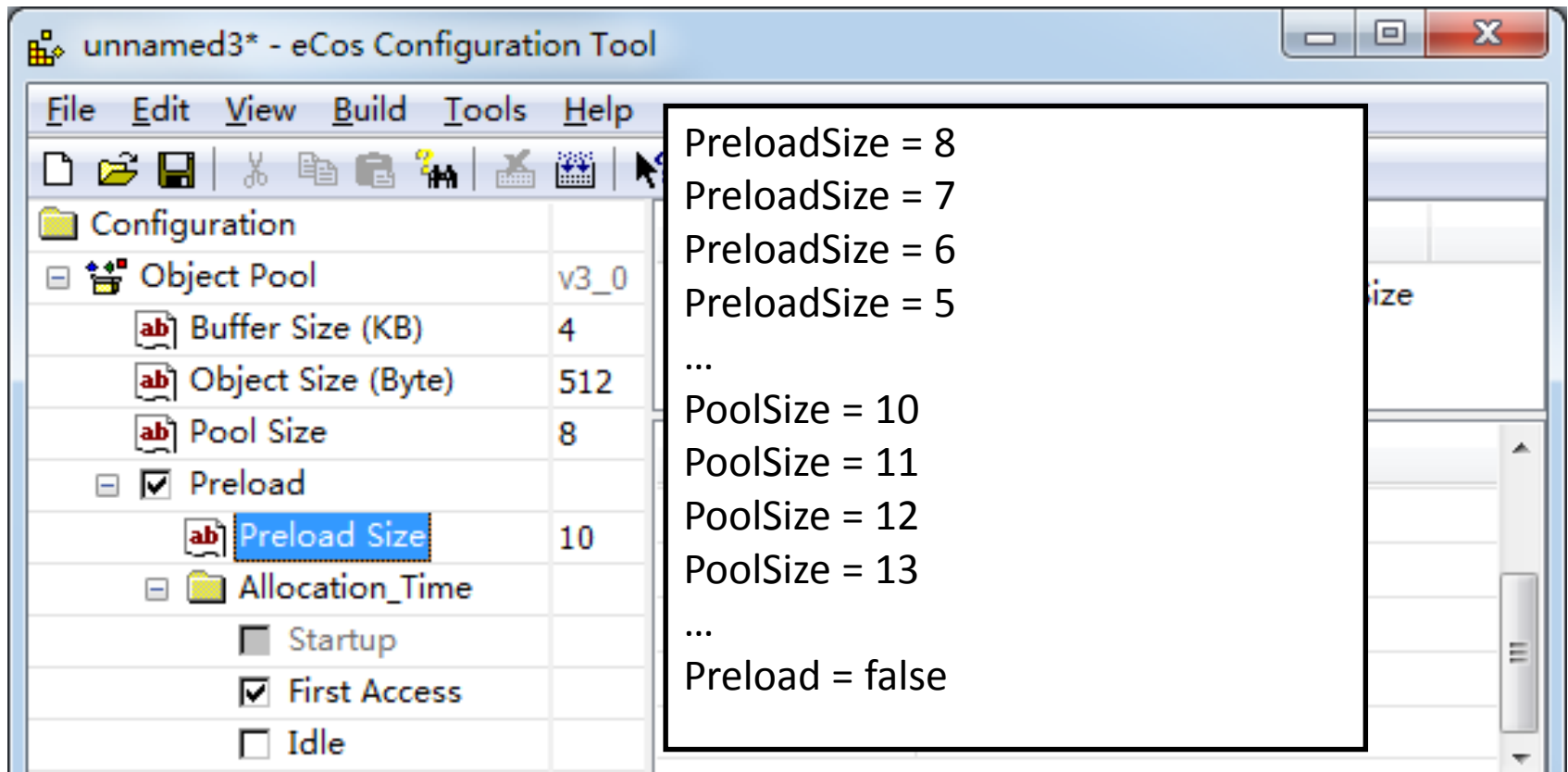
Item	Value
<input checked="" type="checkbox"/> PreloadSize	8

Disable

Further decrease to any value ≤ 8

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

How to complete fixes



Our Solution – Range Fixes

The screenshot shows the 'unnamed3* - eCos Configuration Tool' window. The left pane displays the configuration tree with 'Object Pool' expanded, showing 'Buffer Size (KB)' (4), 'Object Size (Byte)' (512), 'Pool Size' (8), 'Preload' (checked), and 'Preload Size' (10). The right pane shows the 'Preload Size' property with a value of 10. A text box highlights the range fixes: [PreloadSize <= 8], [PoolSize >= 10], and [Preload = false].

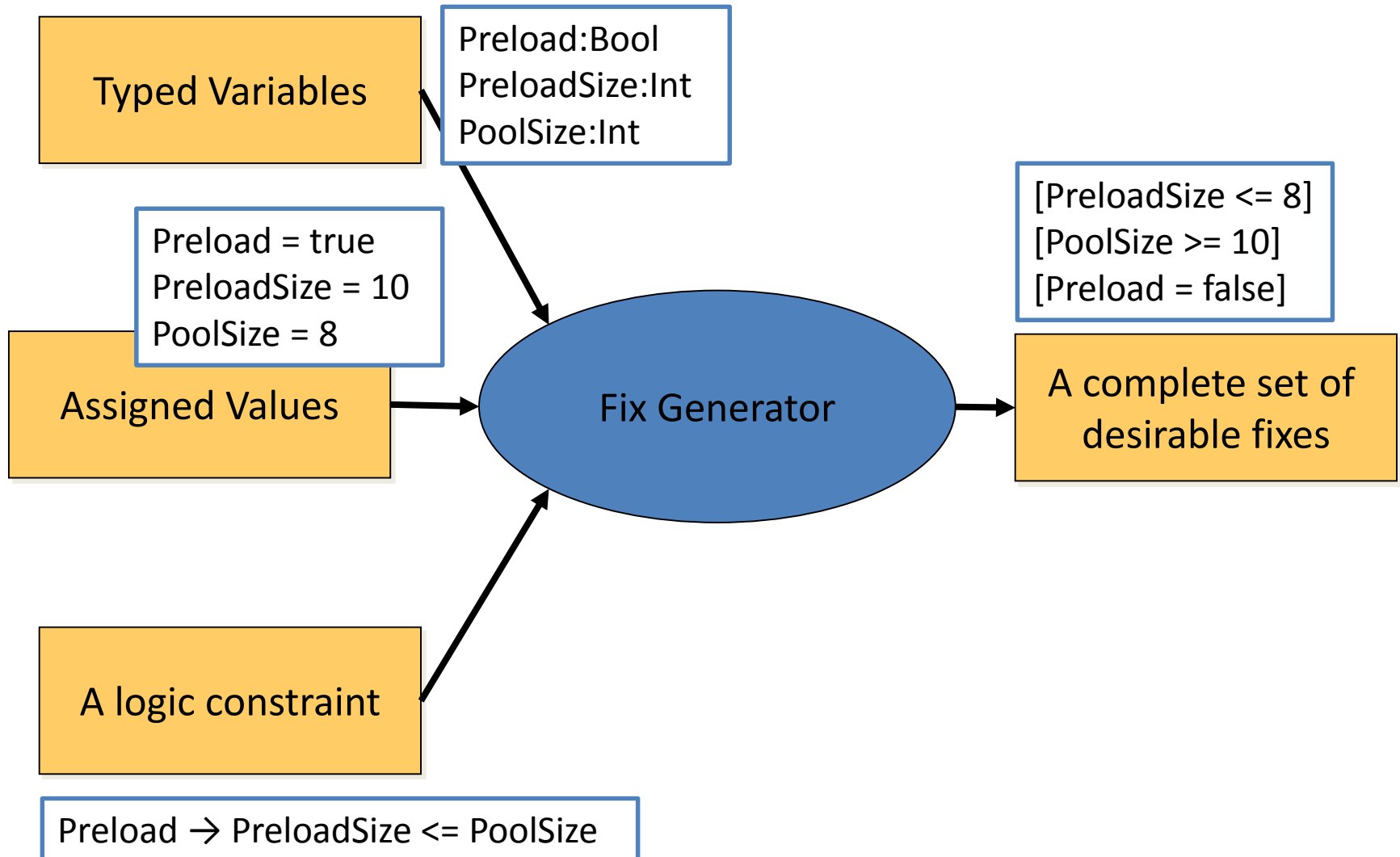
Item	Property
PreloadSize	Requires PreloadSize <= PoolSize
Flavor	data
Requires	PreloadSize <= PoolSize
DefaultValue	10

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]

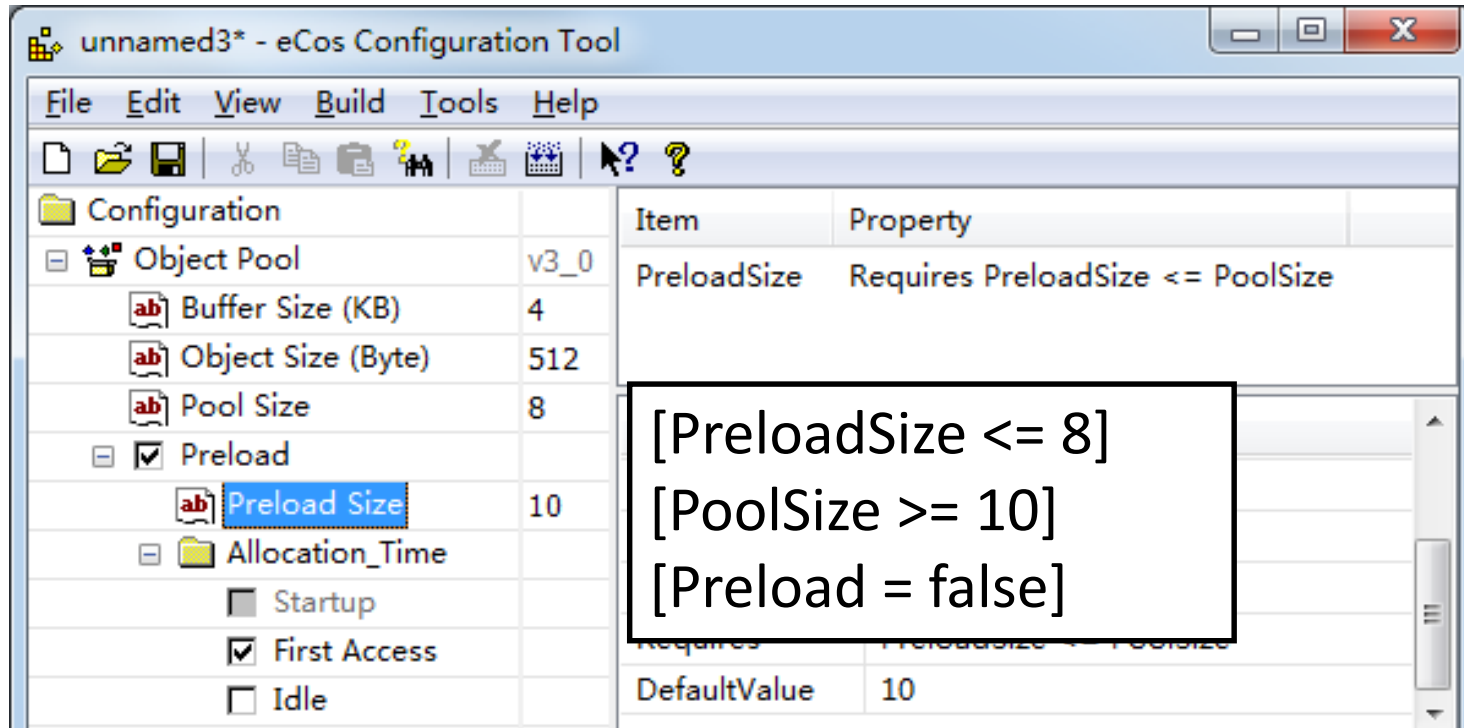
Algorithm Outline

- Step 1: find the variables to change
 - Basic idea: translating to an SMT problem
 - ① treat configurations also as soft constraints
 1. [soft] Preload = true
 2. [soft] PreloadSize = 10
 3. [soft] PoolSize = 8
 4. [hard] Preload \rightarrow PreloadSize \leq PoolSize
 - ② ask an SMT solver for unsatisfiable cores
 - (1, 2, 3)
 - ③ pick one variable from each core
 - {Preload}, {PreloadSize}, {PoolSize}

Algorithm Outline

- Step 2: find the range of the variables
 - Basic idea: simplify the constraint
 - Example: {PreloadSize}
 - ① replace unchangeable variables with their current values
 - $\text{true} \rightarrow \text{PreloadSize} \leq 8$
 - ② simplify the constraint and convert to CNF
 - $[\text{PreloadSize} \leq 8]$

Constraint Interaction



The screenshot shows the 'eCos Configuration Tool' window for 'unnamed3*'. The left pane displays a tree view of configuration items. The right pane shows a table of items and their properties.

Item	Property
PreloadSize	Requires PreloadSize <= PoolSize
DefaultValue	10

Configuration items and values:

- Object Pool: v3_0
 - Buffer Size (KB): 4
 - Object Size (Byte): 512
 - Pool Size: 8
 - Preload: ☒
 - Preload Size: 10
 - Allocation_Time
 - Startup: ☐
 - First Access: ☒
 - Idle: ☐

Constraints shown in the text box:

- [PreloadSize <= 8]
- [PoolSize >= 10]
- [Preload = false]

Constraint Interaction

The screenshot shows the eCos Configuration Tool interface. The left pane displays the configuration tree with 'Object Pool' expanded, showing 'Buffer Size (KB)' at 4, 'Object Size (Byte)' at 512, and 'Pool Size' at 12 (circled in red). A red arrow points from the '12' to a callout box at the bottom left that says 'Increase PoolSize'. The right pane shows a table of properties for the 'PoolSize' item, with a 'Requires' constraint: $\text{PoolSize} == \text{BufferSize} * 1024 / \text{ObjectSize}$. A red arrow points from this constraint to a callout box at the top right that says 'Causing another error'. At the bottom right, another callout box says 'Interacting constraint'.

Causing another error

Configuration Tool

File Edit View Build Tools Help

Configuration

- Object Pool (v3_0)
 - Buffer Size (KB): 4
 - Object Size (Byte): 512
 - Pool Size: 12
 - Preload
 - Preload Size: 10
 - Allocation_Time
 - Startup
 - First Access
 - Idle

Item	Property
PoolSize	Requires PoolSize == BufferSize * 1024 / ObjectSize

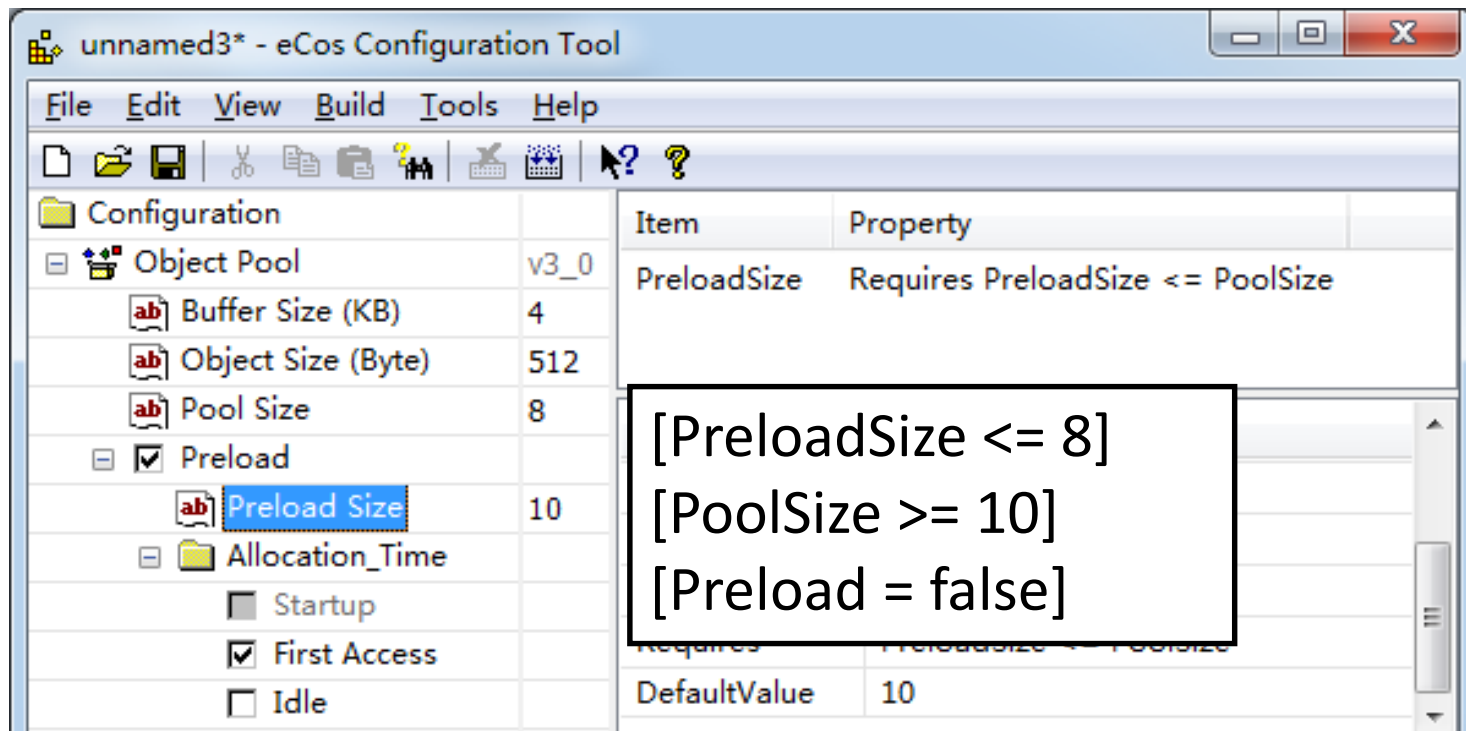
Property	Value
File	unnamed3_install/include/pkgconf/hal.h
Macro	PoolSize
Value	12
Default	0
Flavor	data
Requires	PoolSize == BufferSize * 1024 / ObjectSize

Increase PoolSize

Interacting constraint

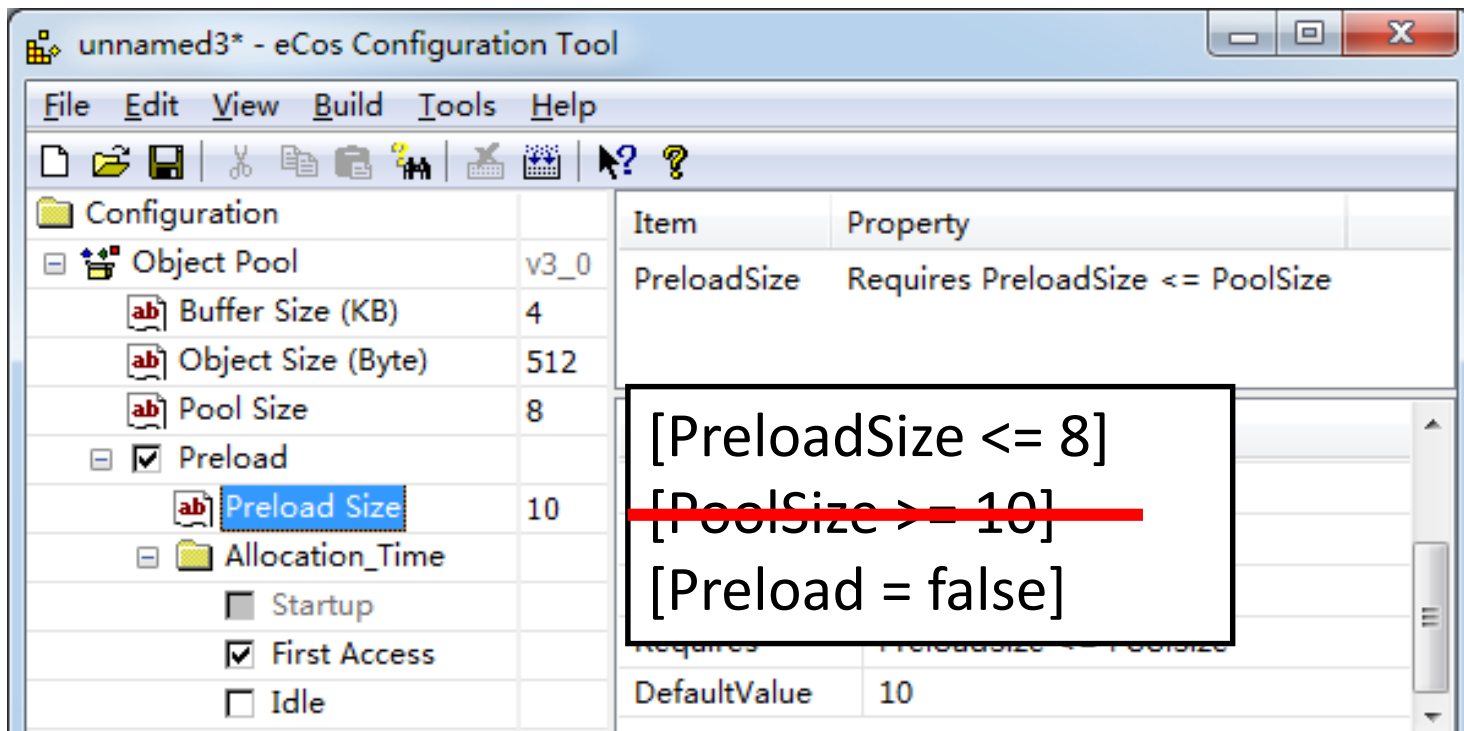
Ignorance

Ignore the interaction



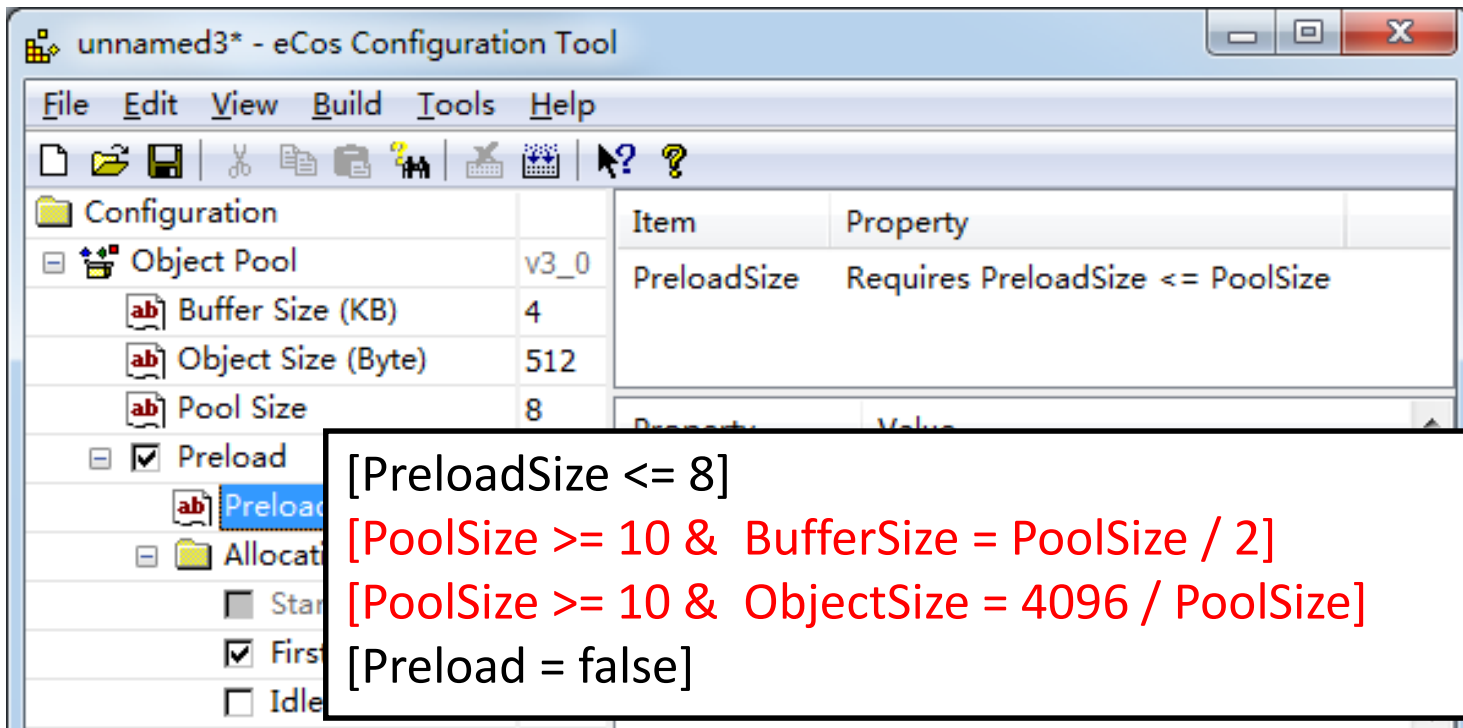
Elimination

Eliminate all changes that will violate other constraints



Propagation

Propagate the change along other constraints



The screenshot shows the 'eCos Configuration Tool' window for 'unnamed3*'. The left pane displays a configuration tree with 'Object Pool' expanded, showing 'Buffer Size (KB)' (4), 'Object Size (Byte)' (512), and 'Pool Size' (8). The 'Preload' checkbox is checked. The right pane shows a table of constraints.

Item	Property
PreloadSize	Requires PreloadSize <= PoolSize

A text box overlay on the bottom right of the screenshot contains the following constraints:

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

Translating to the basic case

- Assignments: Preload = true, PreloadSize = 10, PoolSize = 8, BufferSize = 4, ObjectSize = 512
- Constraints:
 - Preload \rightarrow PreloadSize \leq PoolSize
 - PoolSize == BufferSize * 1024 / ObjectSize
- Ignorance:
 - Preload \rightarrow PreloadSize \leq PoolSize
- Elimination:
 - Preload \rightarrow PreloadSize \leq PoolSize \wedge PoolSize == 4 * 1024 / 512
- Propagation:
 - Preload \rightarrow PreloadSize \leq PoolSize \wedge PoolSize == BufferSize * 1024 / ObjectSize

Comparison of Strategies

	Ignorance	Elimination	Propagation
Execution time	Shortest	Short	Possibly 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	Propagation
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%

Problem: Large Fixes

	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

How to guide the users to
identify their desirable fixes?

Our Solution

- Use the idea of **priority**
 - The priority of a variable **represents the likelihood of its current value being desirable to the user.**
- Two Basic ideas:
 - Generate fixes that only change variables with lower priorities
 - Dynamically adjust the priority of variables through implicit translation of user feedback

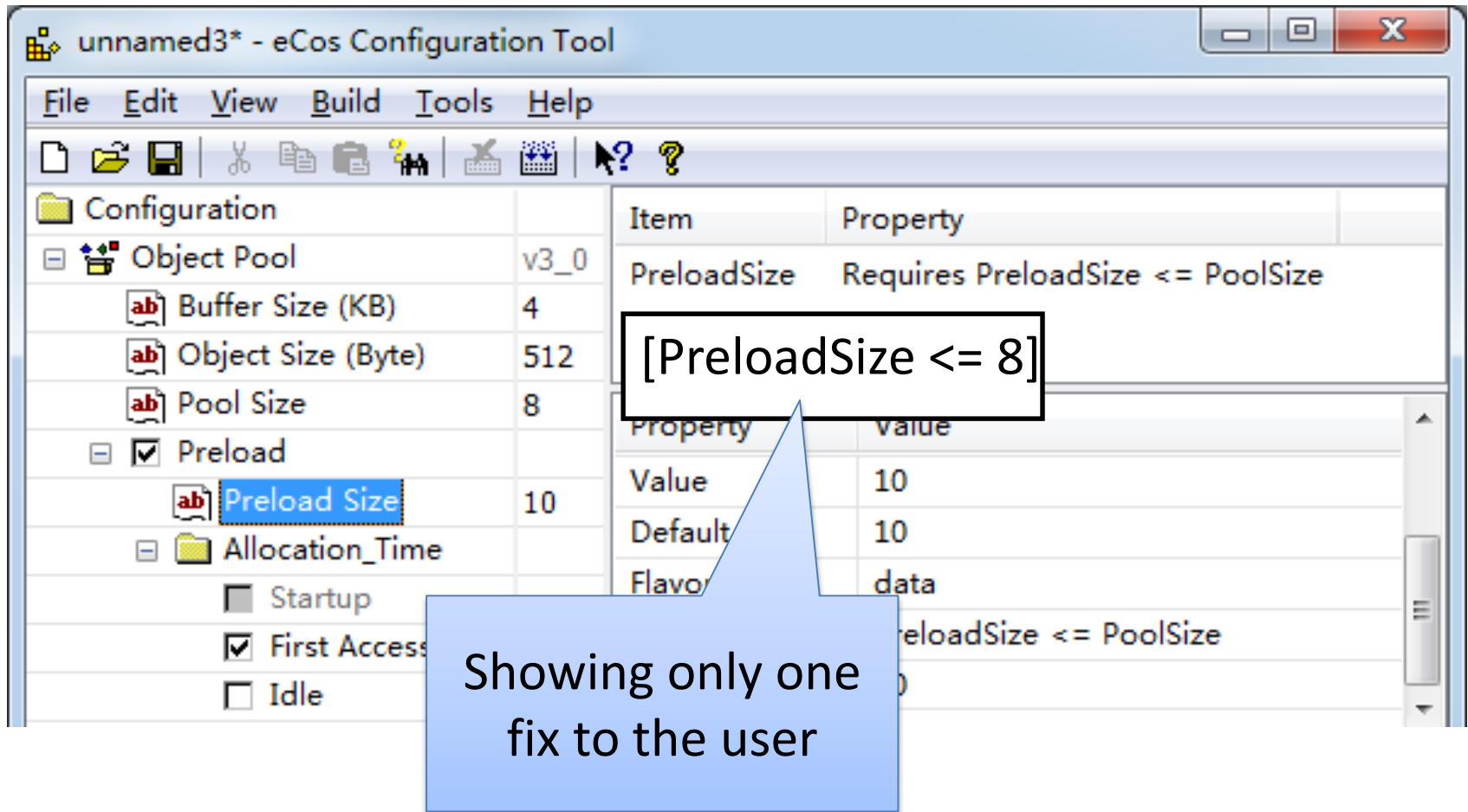
Our Contribution

- A **priority-based** approach to locating a desirable fix through user feedbacks
- An algorithm to implement the approach using **any fix generation algorithm**
- An empirical evaluation that shows the **overall reduction** of choices exposed to the user

Our Contribution

- A **priority-based** approach to locating a desirable fix through user feedbacks
- An algorithm to implement the approach using **any fix generation algorithm**
- An empirical evaluation that shows the **overall reduction** of choices exposed to the user

Our Approach



Provide feedback for each variable

- Accept the change (and pick a value)
- Reject the change
 - *Fix* duration
 - Current range is incorrect, future fixes can propose changes for this variable
 - *Error* duration
 - Current range is correct when fixing this error
 - *Permanent* duration
 - Current range is correct in the whole configuration process

Our Approach

The screenshot shows the 'unnamed3* - eCos Configuration Tool' window. The left pane shows a tree view with 'Object Pool' expanded, showing 'Buffer Size (KB)' (4), 'Object Size (Byte)' (512), 'Pool Size' (8), and 'Preload' (checked). Under 'Preload', 'Preload Size' is highlighted with a value of 10. The right pane shows a table with 'Item' and 'Property' columns. The 'PreloadSize' item has the property 'Requires PreloadSize <= PoolSize'. A callout box points to the 'Preload Size' value of 10, containing the text '[PreloadSize <= 8]'. Another callout box points to the 'Preload Size' row, containing the text 'Reject with Fix Duration'.

Item	Property
PreloadSize	Requires PreloadSize <= PoolSize

[PreloadSize <= 8]

Reject with Fix Duration

Our Approach

The screenshot shows the 'eCos Configuration Tool' window for 'unnamed3*'. The 'Configuration' tree on the left includes 'Object Pool' (v3_0) with properties: 'Buffer Size (KB)' (4), 'Object Size (Byte)' (512), and 'Pool Size' (8). The 'Preload' checkbox is checked. A callout box points to the 'Preload' checkbox with the formula: $[PoolSize \geq 10 \ \& \ BufferSize = PoolSize / 2]$. Below this, two callout boxes provide the logic: 'Accept with PoolSize = 16' and 'Reject with Error Duration'.

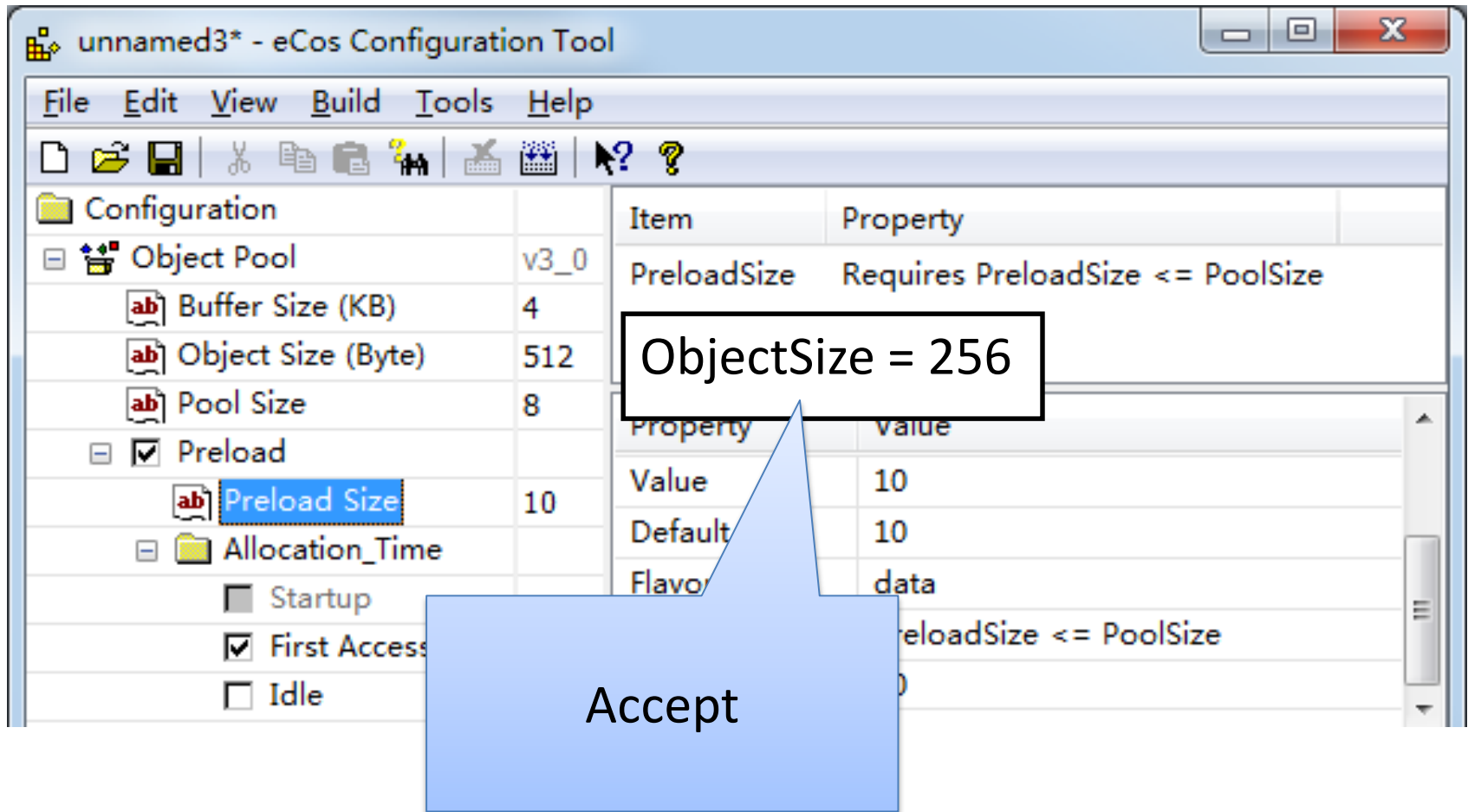
Item	Property
PreloadSize	Requires PreloadSize \leq PoolSize

Property	Value
Preload	<input checked="" type="checkbox"/>

Accept with PoolSize = 16

Reject with Error Duration

Our Approach



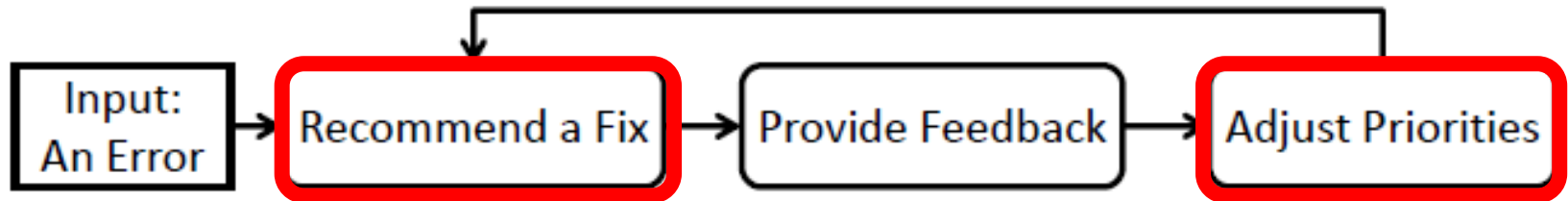
The user feedbacks are stored so that later fixes will be smarter.

Our Contribution

- A **priority-based** approach to locating a desirable fix through user feedbacks
- An algorithm to implement the approach using **any fix generation algorithm**
- An empirical evaluation that shows the **overall reduction** of choices exposed to the user

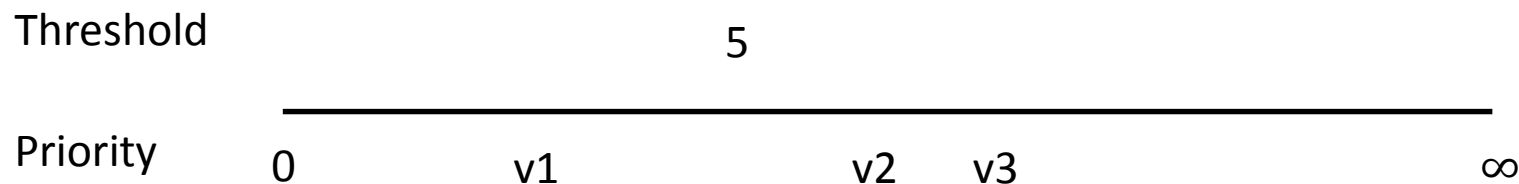
Algorithm Overview

Each variable is assigned a priority, initially zero.



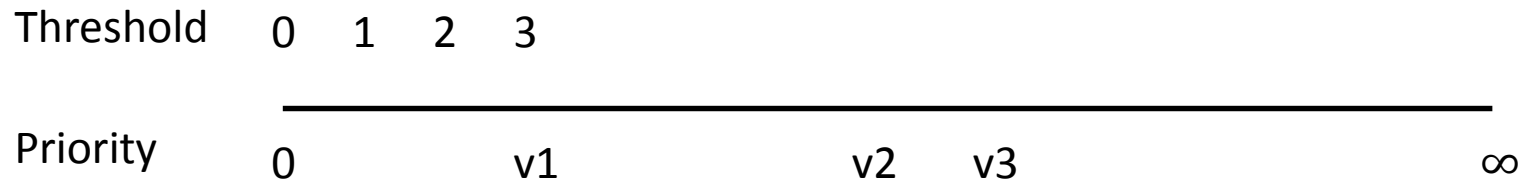
Recommend a fix

- Use a threshold to confine the fix generation scope
 - Variables are changeable only when priority \leq threshold.
 - Constraint [variable = current_value] is added for variables whose priority $>$ threshold



Recommend a fix

- Initial threshold for an error = 1
- Invoke the fix generator
 - Randomly pick one fix from the generated fix list
 - Threshold += 1 if no fix is generated



Adjust Priorities

- New value is assigned
 - priority = 0
- Reject with *Fix* duration
 - priority +=1
- *Reject with Error* duration
 - priority binds to <threshold> +1
 - will be updated when threshold increases
- *Reject with Permanent* duration
 - priority = <max>

Handling No fixes

- Provide users with the variables with *error* and *permanent* durations
- Users should change the durations

Our Contribution

- A **priority-based** approach to locating a desirable fix through user feedbacks
- An algorithm to implement the approach using **any fix generation algorithm**
- An empirical evaluation that shows the **overall reduction** of choices exposed to the user

Supporting Tool: Smart Fixer

1

Configuration	Item	Conflict	Property
Global build options	CYGPKG_HAL_POWERPC_VL...	Unsatisfi...	Requires CYGHWR_HAL_POWERPC_PPC4XX == "405"
Global command prefix	CYGPKG_HAL_POWERPC_VL...	Unsatisfi...	Requires ICYHWR_HAL_POWERPC_ENABLE_MMU
Global compiler flags	CYGSEM_VIRTEX4_GPIO_C...	Unsatisfi...	Requires MNDHWR_VIRTEX4_CHARLCD
Global linker flags	CYGPKG_IO_SERIAL_POWE...	Unsatisfi...	Requires MNDHWR_VIRTEX4_UART
<input checked="" type="checkbox"/> Build GDB stub ROM image	CYGPKG_POSIX_CLOCKS	Unsatisfi...	Requires CYGBLD_ISO_STRUCTTIMEVAL_HEADER == "<cyg/posix/sys/time.h>"
<input type="checkbox"/> Build common GDB stub ROM in	CYGBLD_BUILD_GDB_STUBS	Unsatisfi...	Requires CYG_HAL_STARTUP == "ROM"
Redboot HAL options	CYGBLD_BUILD_GDB_STUBS	Unsatisfi...	Requires CYGSEM_HAL_ROM_MONITOR
Mouse driver for virtex4	CYGBLD_BUILD_GDB_STUBS	Unsatisfi...	Requires CYGBLD_BUILD_COMMON_GDB_STUBS
	CYGBLD_BUILD_GDB_STUBS	Unsatisfi...	Requires CYGDBG_HAL_DEBUG_GDB_INCLUDE_STUBS

2

Fixes

Filter by: All

#	Fix
1	CYGDBG_HAL_DEBUG_GDB_INCLUDE_STUBS_bool:=true CYGDBG_HAL_COMMON_INTERRUPTS_SAVE_MINIMUM_CONTEXT_bool:=false CYGDBG_HAL_COMMON_CONTEXT_SAVE_MINIMUM_bool:=false CYGSEM_HAL_VIRTUAL_VECTOR_CLAIM_COMMS_bool:=true CYGSEM_HAL_USE_ROM_MONITOR_bool:=false

Expand fix

3

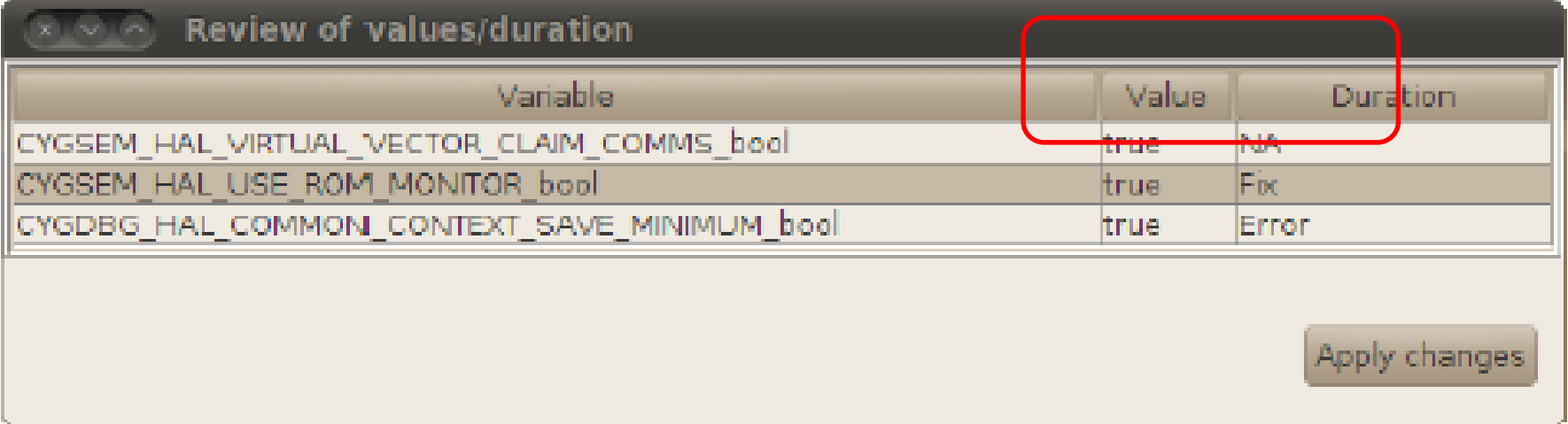
Fix

Select	Fix Unit	Assignment (var = value, ...)	Duration
<input checked="" type="checkbox"/>	CYGDBG_HAL_DEBUG_GDB_INCLUDE_STUBS_bool:=true		NA
<input checked="" type="checkbox"/>	CYGDBG_HAL_COMMON_INTERRUPTS_SAVE_MINIMUM_CONTEXT_bool:=false		NA
<input type="checkbox"/>	CYGDBG_HAL_COMMON_CONTEXT_SAVE_MINIMUM_bool:=false		Error
<input type="checkbox"/>	CYGSEM_HAL_VIRTUAL_VECTOR_CLAIM_COMMS_bool:=true		Error
<input type="checkbox"/>	CYGSEM_HAL_USE_ROM_MONITOR_bool:=false		Error

Select all Reset Apply changes

(a) SmartFixer: Interactive process GUI for fix resolution

Smart Fixer: providing feedbacks



Review of values/duration

Variable	Value	Duration
CYGSEM_HAL_VIRTUAL_VECTOR_CLAIM_COMMS_bool	true	NA
CYGSEM_HAL_USE_ROM_MONITOR_bool	true	Fix
CYGDBG_HAL_COMMON_CONTEXT_SAVE_MINIMUM_bool	true	Error

Apply changes

Evaluation

- Sources
 - Version history from 2 open source projects that cause large fix lists
 - Simulate the user change from the default configurations to the final configurations

Project	Architecture	Variables	Constraints	Errors (initial conf.)
ReconOS	virtex4	933	330	56
	xilinx	765	272	48

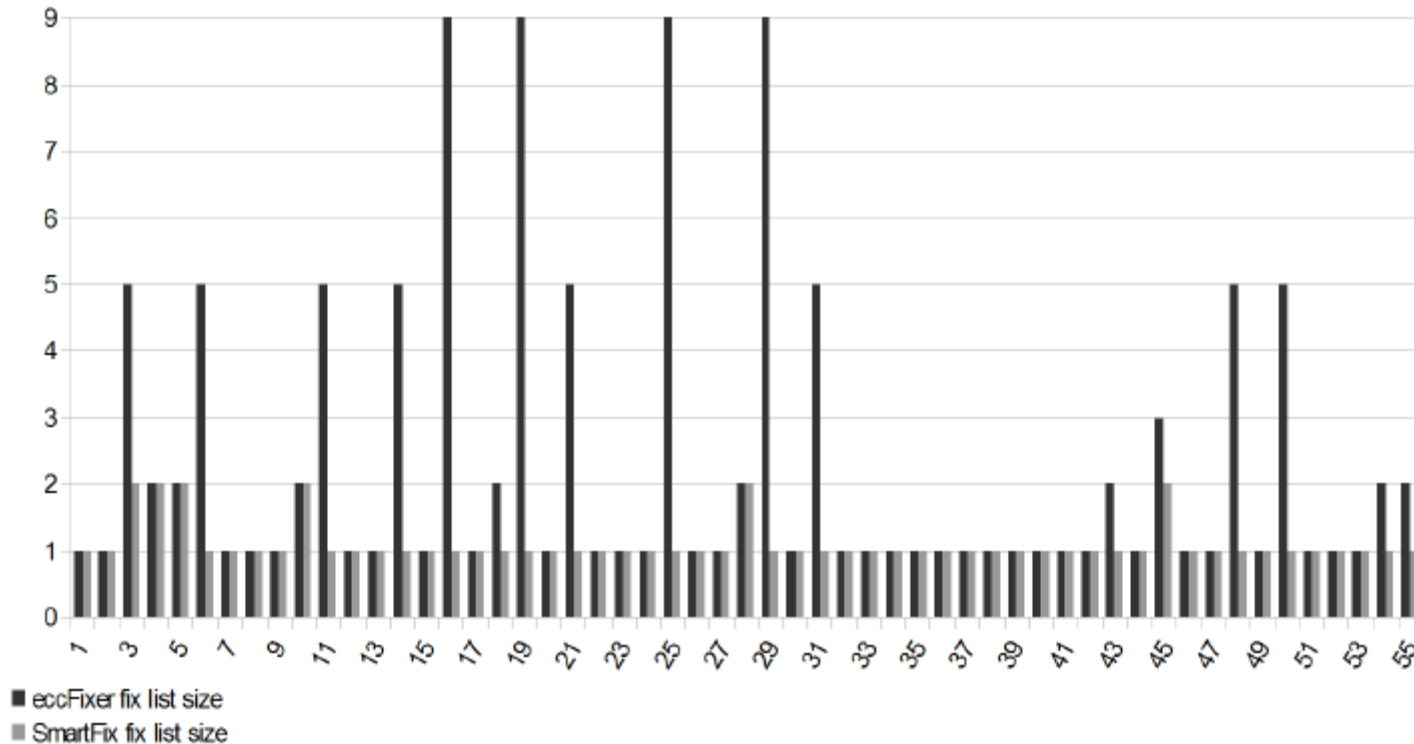
Evaluations

- Steps:
 - Generate a fix for each error, simulate the user feedback

Situation #	Current Value	Fix Changes	Final Value	Operation
1	$a = 1$	$a < 1$	$a = 2$	Reject Fix duration
2	$a = 1$	$a > 1$	$a = 2$	Accept Assign new value
2s	$a = 2$	$a > 2$	$a = 2$	Reject Error duration

- Count the number of fixes and variables

Evaluation Results – virtex4 (1/2)

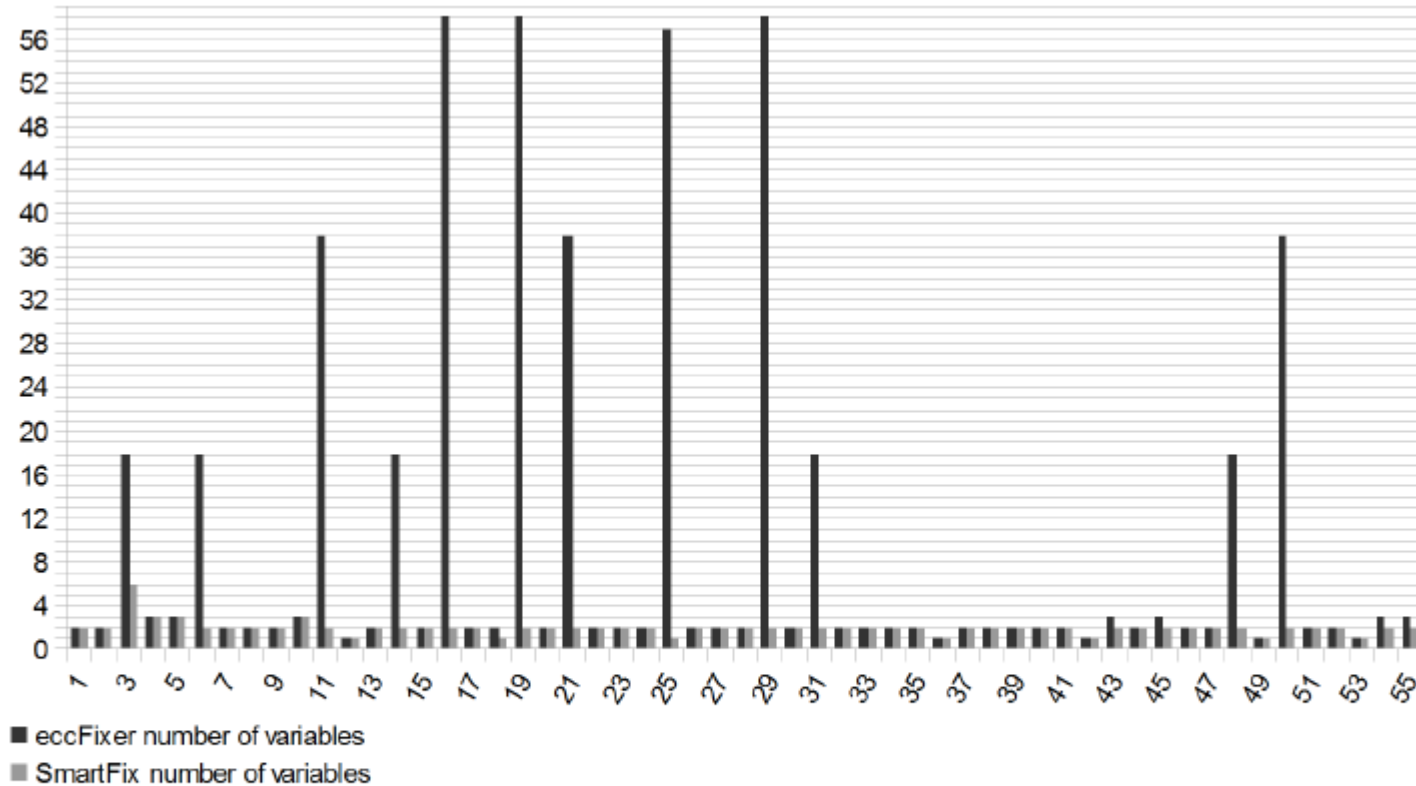


(a) Fix list size

The number of fixes is decreased in 31% of the errors.

In average, there is a reduction of 22%, with a maximum reduction of 89% in the number of fixes

Evaluation Results – virtex4 (2/2)

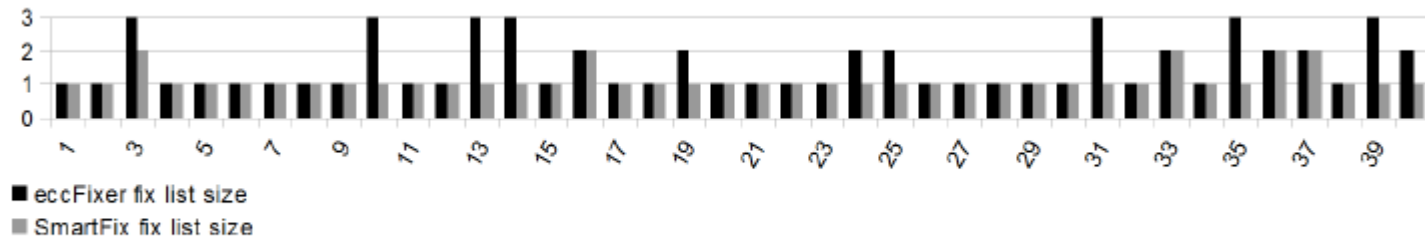


(b) Number of variables

Fig. 6: Experimental results for virtex4

The number of variables is decreased by 23% in average, with a maximum reduction of 98%

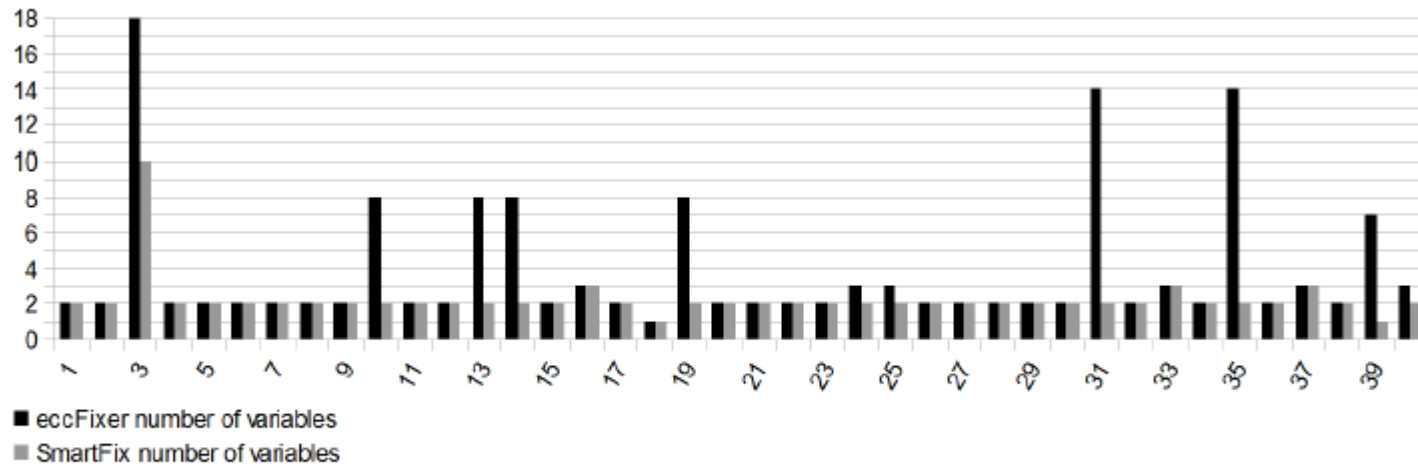
Evaluation Results – xilinx (1/2)



(a) Fix list size

**The number of fixes is decreased in 28% of the errors.
In average, there is a reduction of 16%, with a maximum
reduction of 2/3 in the number of fixes**

Evaluation Results - xilinx (2/2)



(b) Number of variables

**The number of variables is decreased by 18% in average,
with a maximum reduction of 86%**

Summary

- Error Resolution is difficult in configuring large systems
- Range fixes can be generated efficiently
- Large fix list could be controlled by priorities

Thank you for your attention!

- **References:**
- Yingfei Xiong, Arnaud Hubaux, Steven She, Krzysztof Czarnecki. Generating Range Fixes for Software Configuration. In ICSE'12: Proceedings of 34th International Conference on Software Engineering, pages 89-99, June 2012.
- Bo Wang, Leonardo Passos, Yingfei Xiong, Krzysztof Czarnecki, Haiyan Zhao, Wei Zhang. SmartFixer: Fixing Software Configurations based on Self-adaptive Priorities. In SPLC'13: Proceedings of 17th International Software Product Line Conference, August 2013.
- Arnaud Hubaux, Yingfei Xiong, Krzysztof Czarnecki. A User Survey of Configuration Challenges in Linux and eCos. VaMoS'12: Sixth International Workshop on Variability Modelling of Software-intensive Systems, January 2012.
- Leonardo Passos, Marko Novakovic, Yingfei Xiong, Thorsten Berger, Krzysztof Czarnecki, Andrzej Wasowski. A Study of Non-Boolean Constraints in Variability Models of an Embedded Operating System. FOSD'11: 3rd International Workshop on Feature-Oriented Software Development, June 2011.