SMOKE: Simulink Model Obfuscation Keeping structurE

August 2025

Software Engineering Group, University of Bern

McMaster Centre for Software Certification (McSCert)

Contents

1 Introduction

SMOKE: Simulink Model Obfuscation Keeping structurE is a tool that removes, renames, and/or hides various details of a Simulink model in order to hide or remove confidential information. Hiding can be useful for taking screenshots that can be shared or published. Removing, especially removing functionality, can be useful for sharing the model files itself. Such model files may then be studied, modified or published by interested third parties. SMOKE can remove most model aspects, while leaving the model structure intact. The user can decide which aspects to leave intact, and which one's to remove, depending on what is sensitive, and what is fine to share.

Disclaimer: The authors of this tool make no guarantees that all proprietary/confidential information is indeed removed from the Simulink model file. Users should inspect the model to verify that no proprietary/confidential remains according to their needs.

2 How to Use the Tool

This section describes what must be done to setup the SMOKE, as well as how to use the tool.

2.1 Prerequisites and Installation

1. Use MATLAB/Simulink 2024a or newer. It may work with older versions, though.
2. To install the tool, use one of the following approaches to download all necessary files to a location, like `SMOKEdir` of your choice:
   (a) Download the `.zip` files of the SMOKE repository and for the dependency Simulink-Utility. Unzip both at `SMOKEdir`.
   (b) Using a terminal at `SMOKEdir`, clone the SMOKE repository and its dependency:

```
git clone --recursive https://github.com/lanpirot/SMOKE
git clone --recursive https://github.com/McSCert/Simulink-Utility
```

3. Add the `SMOKEdir` of SMOKE and the dependency and all their subfolders to your MATLAB search path. Do this step both for SMOKE and Simulink-Utility! To do this, use one of the following:
   (a) Use the MATLAB current folder panel. Move to the location in the folder panel, then right-click on the folder ⟶ Add to Path ⟶ Selected Folders and Subfolders.
   (b) In the MATLAB console, use `addpath(genpath('SMOKEdir'))`

4. Finally, to keep the path saved for the next MATLAB start, use `savepath` in the MATLAB console.

5. Ensure your Simulink model, you want to obfuscate, is open and unlocked.

Troubleshooting: If running the command "`which SMOKEgui`" indicates that the script is not found, then SMOKE and dependency directories needs to be added to the MATLAB search path. If while executing the program `Unrecognized function or variable 'getInput'` gets printed in the console, then Simulink-Utility and dependency directories needs to be added to the MATLAB search path. For more information on adding files to the MATLAB search path, please see the MathWorks documentation.

2.2 Getting Started

SMOKE can be launched by double-clicking the `SMOKEgui.mlapp` file, or by running the command `SMOKEgui` from the Command Window. This will open the Graphical User Interface (GUI) shown in Figure 1.

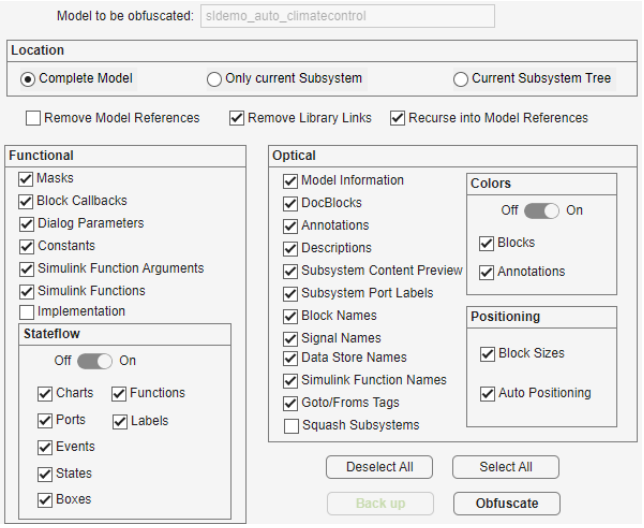

Figure 1: The tool GUI.

2.3 Functionality

This section describes the tool functionality when being used from the GUI (Figure 1). Each section describes one of the sub-panels of the GUI.

2.3.1 Model to be obfuscated

The top line states the currently loaded model (if any). If this is blank, ensure to load a Simulink model. Also make sure, that the model is unlocked, as SMOKE only works on unlocked models.

2.3.2 Scope

Directly below the model's name, you can choose the scope of obfuscation and sanitization. By default the whole model will be affected by the user chosen transformations. You can also restrict SMOKE to only anonymize the current SubSystem's content, or the current SubSystem and every element recursively nested within. This way, you may choose more restrictive anonymizations for some parts of the model, while others may be less obfuscated, or even be left in its original state.

2.3.3 Top Checkboxes

- Model References – The use of a model block introduces a reference to another model. This option resets all model reference blocks so that they no longer point to other models. *Note: This may impact the model functionality.*
- Recurse into Model References – This check-box will apply all changes chosen in Sections 2.3.4, and 2.3.5 inside of any referenced models.
- Library Links – Library links can be used in a model to reference blocks that reside in other libraries. This option removes (or "breaks") all library links so that blocks are stored directly in the model instead of the library. This means that the model is no longer dependant on external libraries.

2.3.4 Left Panel: Functional Sanitization

Here, functionality of the model will be removed, either inside of masks, block callbacks, constant values, stateflow internals, or Simulink Functions. All options leave the model structure intact.

- Masks – Block masks are commonly used to customize the block appearance of custom blocks. This option removes the masks of all blocks. Masks may include special code, that is also removed.
- Block Callbacks – Blocks may have callbacks, that are called on events, like moving them, saving the model etc. They can range from simple (e.g. is another block still present?) to complex.

- Dialog Parameters – Blocks can have all kinds of parameters set. Activate this option to reset them to their default values. This means dialog parameters are a way to store constants in a model.
- Constants – Reset all constants to 1 values.
- Functions – Remove all MATLAB functions from MATLAB function blocks. Inside of a function block, arbitrarily complex functions may be hidden.
- Simulink Function Arguments – A Simulink Function can have inputs and outputs using ArgIn and ArgOut blocks. This option renames the *argument name* of ArgIn and ArgOut blocks to be generic (e.g., u1 for an input, y1 for an output).
- Implementation – Beware: this removes *everything* within the currently selected scope. The model structure will be affected by this.
- Stateflow – These options rename the various named Stateflow elements to have generic names. Currently inputs, outputs, events, boxes, and states are renamed. Transitions are removed.

2.3.5 Right Panel: Optical Obfuscations

The options in the *Right Panel* are of optical nature, only. Use this for screenshots, while leaving the functionality intact.

- Model Information – A Simulink model stores information about itself, such as its creator's name and version number (Figure 2). This option resets this data.
- DocBlocks – A DocBlock stores documentation about the model. This options removes all DocBlocks.
- Annotations – This option deletes all text, area, or image annotations.
- Descriptions – This options removes the *description* information of lines, blocks, and annotations.
- Signal Names – This option turns off signal propagation.
- Subsystem Content Preview – This option turns off the content preview that is displayed in blocks such as subsystems.
- Subsystem Port Labels – This option hides the port labels shown on blocks such as subsystems.
- Block Names – Each block in a model has a *name* that is typically displayed underneath the block. This option renames all block names to a generic name based on the block type. For example, an Inport block will be renamed to Inport1. They are also hidden from view.

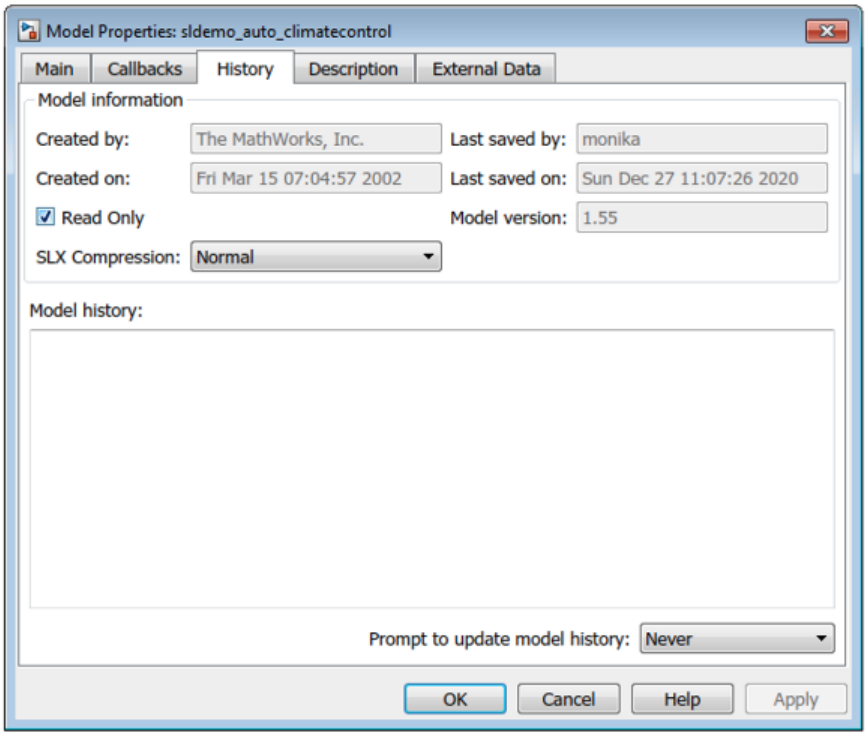

Figure 2: Model Information.

- Signal Names – This option turns off signal propagation.
- Data Store Memory – A Data Store Memory block has a *data store name*. This option renames all Data Store Memory blocks to be generic (e.g., DataStore1) as well as all associated Data Store Read and Data Store Write blocks.
- Simulink Function Names – The trigger within a Simulink Function specifies the function's name. This option renames it to a generic name (e.g., f1), and updates any corresponding Function Caller blocks to match.
- Goto/From Tags – A Goto block has a *goto tag* that matches it to its From blocks. This option renames tags to generic names (e.g., GotoFrom1) and renames any matching From blocks as well.
- Squash Subsystems – Every Subsystem in the scope will be removed. After this obfuscation, all the nested Subsystem content will be presented flatly, without any hierarchy. Note: This will change the model structure.
- Colors – These options remove the colours of blocks and annotations so that they revert to their default color.

- Block Sisez – All block sizes and shapes are reset to their default values.
- Auto Positioning – Simulink's auto positioning function is called. Each call to it may result in a different diagram!

3 Example

Use the command `sldemo_auto_climatecontrol` in the Simulink Command Window to open the example model, shown in Figure 3. To run the tool, run the command `SMOKEgui` from the Command Window, and then press the `Obfuscate` button. The resulting model is given in Figure 4. We can see that the colors, annotations, masks, names, port labels, and many other elements have been removed, renamed, or hidden in the model. Furthermore, all the functionality of the model is removed, because constant values, block parameters, functions are reset and masks and callbacks are removed.

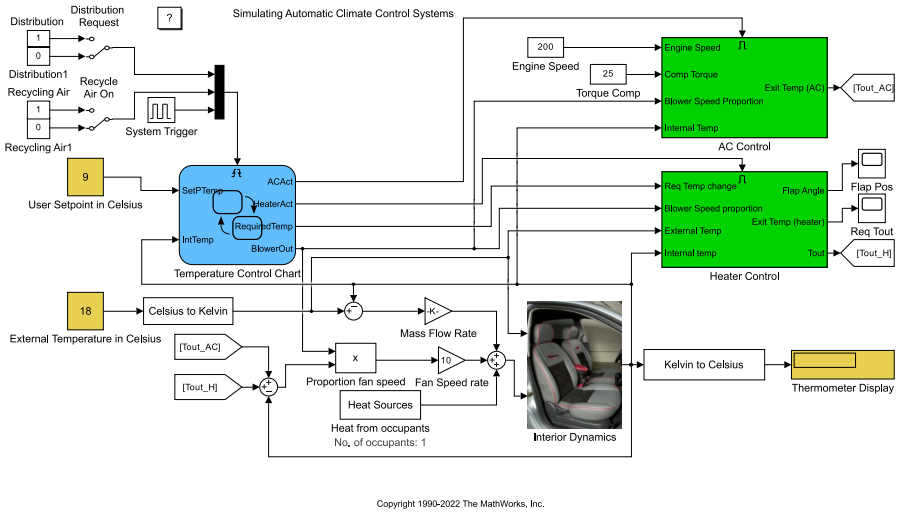

Figure 3: Original demo model.

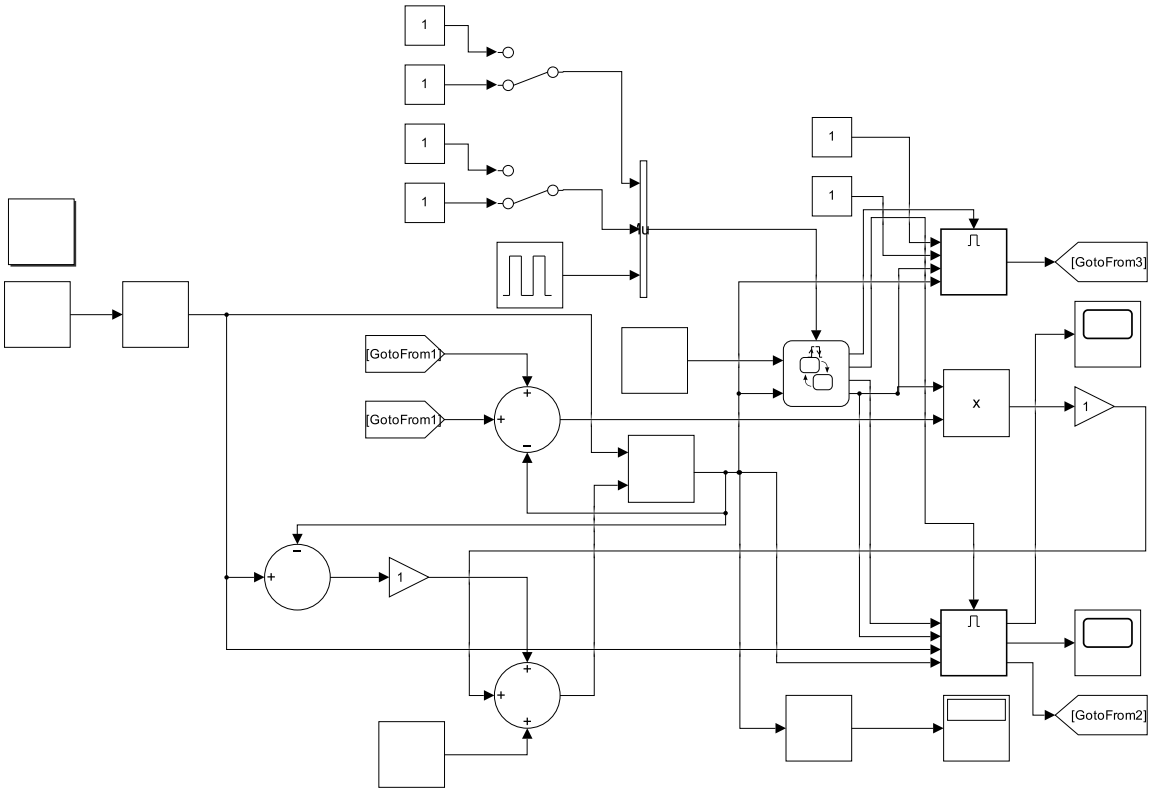

Figure 4: Resulting model after obfuscation.