Logical Simplification for Context Tables in the STPA VS Code Extension

Risk analysis of systems such as cars or aircraft is important to prevent injuries or death. One relatively new technique is System-Theoretic Process Analysis (STPA), which focuses on unsafe interactions between system components. In order to identify such unsafe control actions, a context table is created based on the process model variables of the controller.
Example of a context table:

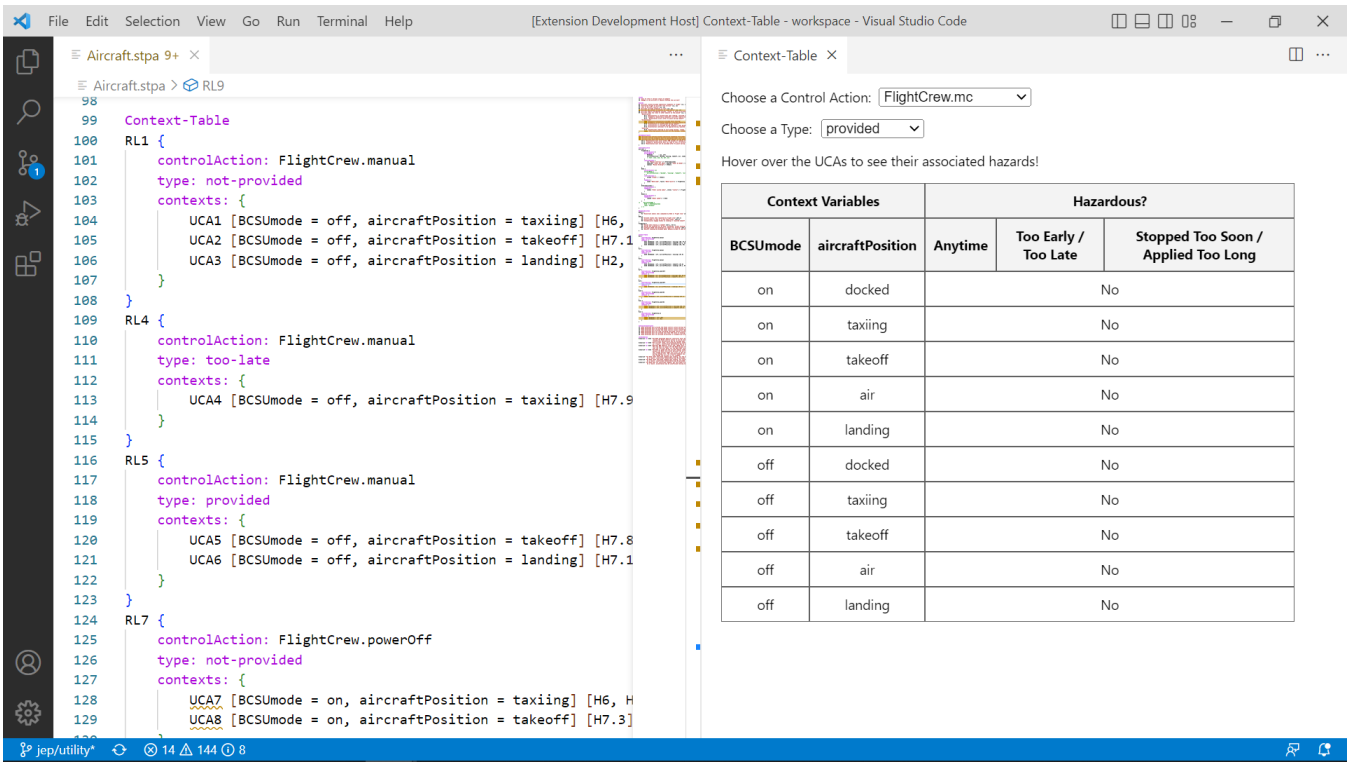
Train Motion	Emergency	Train Position	Hazardous control action?		
			provided anytime	provided too early	provided too late
moving	no emergency	not aligned with platform	Yes	Yes	Yes
moving	no emergency	aligned with platform	Yes	Yes	Yes
moving	emergency exists	not aligned with platform	Yes	Yes	Yes
moving	emergency exists	aligned with platform	Yes	Yes	Yes
stopped	emergency exists	not aligned with platform	No	No	Yes
stopped	emergency exists	aligned with platform	No	No	Yes
stopped	no emergency	not aligned with platform	Yes	Yes	Yes
stopped	no emergency	aligned with platform	No	No	No

The "Hazardous?"-columns are filled out with so called "Rules". The STPA VS Code Extension offers a DSL in which an analysis can be done textually including the defintion of such Rules and the context table is generated automatically. The problem is that the table can get very large with increasing variable numbers. That is where **logical simplification** should help. Its goal is to merge rows where the value of one or more variables can be set to "ANY" because the "Hazardous"-columns are equal. Applying logical simplification to the example leads to the following:

Train Motion	Emergency	Train Position	Hazardous control action?		
			provided anytime	provided too early	provided too late
moving	no emergency	any	Yes	Yes	Yes
moving	emergency exists	any	Yes	Yes	Yes
stopped	emergency exists	any	No	No	Yes
stopped	no emergency	not aligned with platform	Yes	Yes	Yes
stopped	no emergency	aligned with platform	No	No	No

The goal of this thesis is to find ways to determine which rows of a given context table can be merged and compare the complexity.

STPA VS Code Extension



(kieler/stpa (github.com))

Goals

- Find ways to determine which rows of a given context table can be merged
- Compare the complexity of the approaches in regard to time and memory
- Optional: implement the best approach

Scope

Master's Thesis

Related Work/Literature

- J. Petzold, *A Textual Domain Specific Language for System-Theoretic Process Analysis*. Master Thesis, Department of Computer Science, Kiel University, 2022. (https://rtsys.informatik.uni-kiel.de/~biblio/downloads/theses/jet-mt.pdf)
- J. P. Thomas, *Extending and automating a systems-theoretic hazard analysis for requirements generation and analysis*. Diss, Massachusetts Institute of Technology, 2013. (https://dspace.mit.edu/bitstream/handle/1721.1/81055/857791969-MIT.pdf?sequence=2&isAllowed=y) Chapter 3.3, 5.2
- E. J. McCluskey, *Minimization of Boolean functions*. The Bell System Technical Journal, 1956. (https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6769983)

Involved Languages/Technologies

- TypeScript (https://www.typescriptlang.org/)
- VS Code API (https://code.visualstudio.com/api/references/vscode-api)

Supervised by

Jette Petzold
jep@informatik.uni-kiel.de