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:

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

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 definition 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:

	Emergency		Hazardous control action?				
Train Motion		Train Position	provided	provided too	provided too		
			anytime	early	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

× F	ile Edit	Sele	ction View Go Run Terminal Help [Extension Development Host]	Co	ntext-Table - wo	rkspace - Visual Stud	io Code		□□□0: -	đ	×
Ŋ	E Aircr	aft.stp	a 9+ × ····		E Context-Tabl	le ×					□ …
	≡ Airc	E Aircraft.stpa > ♀ RL9			Choose a Control Action: FlightCrew.mc						
\mathcal{Q}	99	99 Context-Table									
~	100	RL1	{	1120200	choose a type. provided						
90 01	101 controlAction: FlightCrew.manual				Hover over the UCAs to see their associated hazards!						
	102	102 type: not-provided								1	
₹ S	103		contexts: {		Context Variables		Hazardous?				
	104		UCA1 [BCSUmode = off, aircraftPosition = taxiing] [H6, Theatmace.					Too Farby (Channed Tee Coon (on (1
	105		UCA2 [BCSUmode = off, aircraftPosition = takeoff] [H7.1 VMTCharmon		BCSUmode	aircraftPosition	Anytime	Too Late	Applied Too Long	ona ona	
E B	106		UCA3 [BCSUmode = off, aircraftPosition = landing] [H2,					100 Lute	Applied 100 E	Jing	4
	107	1			on	docked	No				
	109	RL4			00	taviing			No		1
	110		controlAction: FlightCrew.manual			taxiirig				_	
	111		type: too-late		on	takeoff			No		
	112		contexts: {								
	113		UCA4 [BCSUmode = off, aircraftPosition = taxiing] [H7.9		on	air	No				
	114		}		on	landing]
	115	}				landing					-
	116	RL5	•		off	docked	No				
	117		controlAction: FlightCrew.manual								-
	118		type: provided		off	taxiing			No		
	119		CONTEXTS: {		off	takeoff	No		No]
	121		UCA6 [BCSUmode = off, aircraftPosition = landing] [H7.1								-
	122		}		off air				No		
	123	}			off	landing	No				
	124	RL7	{								_
	125		controlAction: FlightCrew.powerOff								
(8)	126		type: not-provided								
	127		contexts: {								
	128		UCAX [BCSUmode = on, arrcrattPosition = taxling] [H6, H UCA8 [BCSUmode = on, aircraftPosition = takeoff] [H7.3]								
											ም 🗘

(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