Appendix: The Slow and The Furious? Performance Antipattern Detection in Cyber-Physical Systems

Imara van Dinten[a,*], Pouria Derakhshanfar[b], Annibale Panichella[a], Andy Zaidman[a]

[a] *Delft University of Technology, Van Mourik Broekmanweg 6, 2628 XE, Delft, The Netherlands*
[b] *JetBrains Research, Huidekoperstraat 26-28, 1017 ZM, Amsterdam, The Netherlands*

This appendix consists of additional tables and information for Study I of the paper "The Slow and The Furious? Performance Antipattern Detection in Cyber-Physical Systems".

List of Tables

1. Initial Keyword Selection

This section relates to the paper's section 3.2.1 "Initial Keyword Selection".

Shown in Table 1 are the keywords selected for the Initial Keyword set, including an explanation for each sub-group. These keywords are also validated with domain experts from the H2020 COSMOS project [1].

From the Initial Keyword Set, the number of occurrences of each keyword is visualized in Table 2. Further, shown in this table is the number of instances a commit containing this keyword also contained an antipattern.

*Corresponding author
Email addresses: I.vanDinten@tudelft.nl (Imara van Dinten), A.Panichella@tudelft.nl (Annibale Panichella)

Keywords	Explanation
performance, runtime	As the focus of the research is performance, the keywords 'performance' and 'runtime' link directly to any commit that is related to this area.
slow, slower, slowing, fast, faster, increase, decrease	These adjectives are used to indicate a change in the commit in the described way. This could indicate a performance improvement or decrease.
memory, memory-heap, memory-leak, memory leak, bottleneck, overhead, deadlock, livelock, infinite, speed, impasse, hang, stuck	These keywords are chosen based on previous experience, books regarding performance and found during the analysis phase.

Table 1: Keywords Description

We decided to share this information in the appendix as this might be of interest as a motivation for future research towards a link between keywords and antipatterns.

2. Keyword Set Expansion with Information Retrieval and Topic Modeling

This section relates to the paper's section 3.2.2 "Keyword Set Expansion with Information Retrieval and Topic Modeling". As decribed in the paper, we applied Information Retrieval (IR) and Topic Modeling (TM) techniques to expand our set of keywords.

As an example, three topics from the PX4-Autopilot project are shown in Table 3. From these three topics, the words *size* and *rate* have been chosen. The full analysis of each project is publically available on GitHub [1]. The TM process resulted in a selection of keywords, for each project, as shown in Table 4.

References

[1] Cosmos: Devops for complex cyber-physical systems, `https://www.cosmos-devops.org/` (2021).

[1] `https://github.com/ciselab/CPS_repo_mining/tree/main/T-Model`

Keyword	Occurrences		Antipatterns	
	Total	**First**	**None**	**Detected**
increase	481	398	115	283
memory	239	230	154	76
performance	101	98	58	40
faster	84	65	29	36
fast	74	57	39	18
slow	69	57	32	25
runtime	50	46	42	4
decrease	53	33	11	22
hang	144	19	17	2
infinite	17	15	11	4
overhead	28	14	8	6
slower	15	12	6	6
deadlock	10	9	3	6
speed	60	1	0	1
memory leak	36	1	1	0
increases	5	1	0	1
stuck	3	1	1	0
slowing	2	2	2	0
bottleneck	3	0	-	-
memory-heap	0	0	-	-
memory-leak	0	0	-	-
livelock	0	0	-	-
impasse	0	0	-	-
Total	**1474**	**1059**	**529**	**530**

Table 2: From the Initial Keyword Set: the number of occurrence of each keyword and instances a commit containing this keyword also contained a antipattern.

	Topic 1	Topic 2	Topic 3
1	param	stack	control
2	config	log	mode
3	default	**size**	attitude
4	romfs	reduce	fw
5	startup	logger	vtol
6	parameters	sdlog	controller
7	params	output	pos
8	board	logging	**rate**
9	configs	px	manual
10	script	buffer	land
11	start	usage	offboard
12	trigger	adjust	att
13	info	queue	transition
14	exclude	start	throttle
15	rcs	debug	tecs
16	meta	space	detector
17	airframe	verbose	parameter
18	scripts	dataman	estimator
19	vtol	priority	altitude
20	camera	ensure	wing

Table 3: Example of Topic Modelling from the PX4-Autopilot project, highlighted are the words selected.

Projects	**Words** *(Nr. of Antipatterns/Nr. of Commits)*
PX4-Autopilot	size *(26/277)*, rate *(50/355)*, speed *(7/175)*, timeout *(38/134)*, unneeded *(1/62)*
Andruino-esp32	corrected *(1/4)*, optimize *(0/7)*, wrong *(0/18)*, size *(0/30)*, timeout *(3/15)*, timer *(0/16)*, iot *(0/11)*
Grbl	cleaned *(2/16)*, reflect *(0/7)*, caused *(0/2)*, changed *(0/10)*, error *(0/24)*, refactored *(0/12)*, override *(0/6)*, cleanup *(0/8)*, tidying *(0/2)*, optimization *(0/7)*, disabling *(0/4)*, size *(0/10)*, edit *(0/17)*, edits *(0/6)*, speed *(0/11)*, improved *(0/9)*, rate *(1/17)*, computation *(0/2)*, clean *(1/29)*, tweaks *(0/5)*, optimized *(1/5)*, corrected *(0/6)*, realtime *(0/2)*
Dronekit Android	cleaned *(0/24)*, broken *(0/11)*, speed *(2/26)*
Node AR Drone	missing *(0/4)*, errors *(0/6)*, simplify *(0/2)*, timeout *(1/4)*, drop *(0/4)*, explicit *(0/2)*, expose *(0/3)*, size *(0/4)*, pave *(0/5)*, wip *(0/2)*, increased *(1/2)*, broken *(0/2)*, fixes *(0/5)*, failing *(0/2)*, check *(0/4)*
Android App Manager	improved *(0/3)*, failure *(0/2)*
Cylon	halting *(0/3)*, improve *(0/17)*
Johnny Five	clock *(0/6)*, scaling *(2/8)*, duration *(0/5)*, improved *(1/17)*, nitpicking *(0/11)*, sensitivity *(1/12)*, time *(3/56)*, speed *(3/20)*
Robonomics-JS	-
Robonomics-Contracts	cleaning *(0/2)*, expenses *(0/3)*
Vacuum Robot Mark II	unnecessary *(0/2)*
TurtleBot	convenient *(0/2)*, cleanup *(0/13)*, compatibility *(0/4)*
TurtleBot 3	compatibility *(0/4)*, rate *(9/10)*, inertia *(1/4)*, frequency *(4/4)*, size *(2/2)*, limit *(1/3)*
Valetudo	time *(1/35)*, timers *(0/9)*, speed *(0/14)*, timer *(0/16)*, improve *(0/19)*

Table 4: Resulting the Keyword Expansion, shown are the additional set of keywords per project.