Towards the Deployment of Knowledge Based Systems in Safety-Critical Systems

De-Grancey Florence[1] and Audouy Amandine[1]

[1] Thales AVS France, SAS, France
firstname.lastname@fr.thalesgroup.com

Abstract. At Thales, we studied the use of Knowledge-Based System (KBS) to create a crew-assistant, inserted inside the safety-critical cockpit systems. Developing a KBS as a safety-critical system induced new needs such as a high amount of verification activities or a bounded reasoning time. This paper aims at presenting our needs and related new challenges to the scientific community.

Keywords: safety-critical system, ontology, knowledge-based system

1 Introduction

When you get on a plane or charge your smartphone with electricity produced by nuclear plants, you rely on safety-critical systems (SCS). Development of such high reliable systems must provide evidences that the system performs well its intended function and does not have any undesirable behavior that could lead to human injury or environmental damages. Since a few years the introduction of AI's technics in SCS is extensively studied, in particular Machine Learning. Knowledge-Based Systems (KBS) are also considered for information retrieval or reasoning-based decision-making tasks. Their ability to add value to existing domain knowledge or to provide a causal explanation makes them attractive.

At Thales, we considered the application of KBS to create a cockpit-assistant supporting crew decision-making during normal and abnormal situation. As example, it should detect automatically if aircraft's landing airport becomes unreachable, explain the causes to the crew, and suggest diversion airports. This assistant relies on an OWL DL ontology where the TBox represent domain knowledge and the ABox represents current situation. Contextualized assistance or suggestions are built in soft real time using a combination of standard reasoning tasks: a *"query task*" extracting implicit information, a *"consistency task"* which uses consistency checks to assess if current situation status is correct, an "*explanation task*" which provides a comprehensible explanations to the crew if errors exists inside ABox, followed by *"root-cause task*" which identifies a way to correct errors. These tasks are based on the Hermit reasonner[1].
Application in a safety-critical system context unveils new requirements concerning knowledge-based technologies that, to our best knowledge, are open scientific challenges. In this paper, illustrating by our results, we want to highlight three of them.

[1] HermiT Reasoner: Home (https://www.hermit-reasoner.com)

2 Facilitating the design of a task-fit knowledge base

Context: Safety-critical systems traditionally follow a certification process that ensure that enough evidences are collected to demonstrate the trustworthiness of the system. KBS introduce a novelty: demonstrating that the knowledge base design is *task-fitted*, meaning that it contains necessary and sufficient elements to perform the intended function inside the desired operational domain, and no more.

Our work: To guide the knowledge base design, we used seminal work of [6] as a guidance: starting from aeronautics domain ontology SESAR BEST[2] AIRM, we refined them to select only the necessary elements for the expected tasks. During design, as proposed in [3], we performed error checking using OntoDebug[3] and monitored metric using OntoMetrics[4]. Metrics of the original and obtained ontology are given in table 1.

Remaining challenge: During ontology design, we noted that selecting pertinent concepts and relations required a high expertise in KB design, making difficult KB assessment. To accelerate the conception of KB, some **methods to help** traditional software engineers would be appreciable. One valuable track could be to **facilitating the assessment of ontology high-level properties** described in [6] using **ontology metrics**. For example, if minimizing the *depth*, can easily be understood as a contributor to ontology *intelligibility* (explanations less complex) or d*eployability* (paths explored quickly), impact of *tangledness* or *attribute richness* is hardly understandable.

Ontology	Classes	Axioms	Richness	Depth	Breadth	Tangledness	Path Nb
BEST	1177	34576	0.167	8	135	0.403	2256
our	97	5989	0.1857	5	14	0.103	127

Table 1. Ontologies Metrics

3 Enhancing black box verification & validation

Context: To allow deployment of KBS in SCS, in addition to classical verification and validation (VV) practices, one must verify that the whole KBS (eg. knowledge base, reasoner and additional algorithm) performs well its intended function and does not present any unintended behaviour. A property of determinism (e.g. same output is obtained for same input), is also required. These verifications can be performed either by testing (called "*black box testing*" in [4]) either by formal demonstration.

Our work: We explored black box testing methods, designing several test-based campaigns where a test is defined by a manually created ABox or a set of variations around theses ABox. Especially for *"consistency-tasks", "root-cause-tasks",* and *"explanation task"* tests, we introduced erroneous axioms leading to ABox inconsistency. In ours campaigns, we defined an *accuracy* criteria as for "*query task*", the percentage

[2] https://www.project-best.eu/
[3] http://isbi.aau.at/ontodebug/
[4] https://ontometrics.informatik.uni-rostock.de/ontologymetrics/

of right answers provided for any possible query on an A-box element; for *"consistency tasks"*, the percentage of detected inconsistency in test cases; for *"root-cause tasks"* and *"explanation task"*, the quantity of tests where algorithm provides the right and same root cause or explanation. As expected, we effectively verified that the KBS provided always the expected output and has deterministic outputs.

Remaining challenges: Even if black-box test-based campaigns provide confidence elements, they do not ensure that the system would perform its intended function **whatever** the ABox e.g. operational conditions. Indeed, as test sets are manually created, we are not able to prove that **every possible ABox** was tested. For low criticality task, we can consider automatic test generation as proposed in [1] but – to our best knowledge – there is no tools to generate automatically inconsistencies in ontology. Developing new tools managing inconsistency would be valuable.

For high criticality tasks, a formal demonstration approach, based on the decidability criteria would be preferable. Developing a tool to demonstrate decidability would be valuable. Furthermore, as to our best knowledge, not all description logics currently satisfy these criteria, proving decidability of new logics would be pertinent.

4 An acceptable and bounded execution time

Context: In typical SCS, outputs must be provided with a guaranteed accuracy and within an acceptable and bounded execution time. For KBS, this requirement translates into the ability to *finish the reasoning task* whatever the A-Box filling, within an acceptable time and without saturating the selected hardware.

Our work: During the test-based campaigns detailed above, we launched each test at least 100 times to measure execution time statistics. We notice a high variability of the execution time for all tasks except query, sometimes leading to execution timeout with respect to specifications. Results are illustrated figure 1.

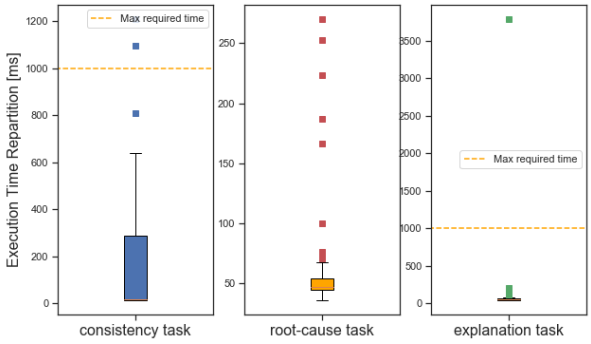

Fig. 1. Repartition of measured execution time and required execution time (1000ms)

Remaining challenges: The variability and the excess of execution time is currently not acceptable. It's well known that OWL DL reasoners have difficulties to scale on large ontology, over than 1000 axioms ([11], [3]) and our ontology clearly overtakes

this limit. **Scaling, accelerating and bounding the execution time of reasoning task** is then a crucial point to enable the deployment of our crew assistant. We note that several interesting tracks are currently studied by scientific community such creating an optimized reasoner ([8], [9], [5]), incremental reasoning [2], parallel reasoning [7] or reasonner composition [10]. They will be explored in additional work.

5 Conclusion

In Thales, we develop a knowledge-based system (KBS) to address the needs of crew assistance in complex and critical situations. Even if the technology presents promising capacities, we are confronted to important challenges before a concrete deployment. Increasing the amount of confidence elements collectable during VV is the first one and concerns both evaluation of the knowledge base itself and the whole KBS. The second challenge is the reasoning acceleration to reach an acceptable and bounded execution time for large ontologies. We encourage the semantic web community to seize these challenges in the next years to enable the deployment of such systems.

Acknowledgements: We thanks V. Charpenay, C. Rey and F. Toumani for their valuable comments and inspiring discussions.

References

1. Banerjee, S., Debnath, N.C. & Sarkar, A.: An Ontology-Based Approach to Automated Test Case Generation. SN Computer Science, 2, 35 (2021).
2. Bento, A., Médini, L., Singh, K., Laforest, F.: Do Arduinos Dream of Efficient Reasoners?. In: The Semantic Web. ESWC 2022. (2022).
3. Bobed, C., Yus, R., & Bobillo, F., & Mena, E.: Semantic Reasoning on Mobile Devices: Do Androids Dream of Efficient Reasoners?. Journal of Web Semantics (2015).
4. McDaniel, M., Storey, V.: Evaluating Domain Ontologies: Clarification, Classification, and Challenges. ACM Computing Surveys. (2019).
5. Motik, B., Horrocks, I., Kim, S. M.: Delta-reasoner: A Semantic Web reasoner for an intelligent mobile platform. In: 21st World Wide Web Conference (WWW2012) (2012).
6. Vizedom, A., Neuhaus, F. and al: Toward Ontology Evaluation across the lifecycle. Applied ontology. (2013).
7. Steigmiller, A., Glimm, B.: Parallelized ABox Reasoning and Query Answering with Expressive Description Logics. In: The Semantic Web. ESWC 2021. (2021).
8. Sinner, A. Kleemann,T.: KRHyper In your pocket, In: 20th Intl. Conf. on Automated Deduction (CADE-20), Vol. 3632, LNCS, Springer, 2005, pp. 45. (2005).
9. Steller, L., Krishnaswamy,S., Gaber,. M. M.: Enabling scalable semantic reasoning for mobile services, International Journal on Semantic Web and Information Systems (2009)
10. Tai, W., Keeney, J., O'Sullivan, D.: Resource-Constrained Reasoning Using a Reasoner Composition Approach. Semantic Web. 6. p35-59. (2015).
11. Pan, Z.: Benchmarking DL reasoners using realistic ontologies. In: Proc. of the workshop on OWL: Experiences and Directions (OWLED 2005), Vol. 188, CEURWS, (2005).