The OWL Reasoner Evaluation Goes Mobile

Roberto Yus, Fernando Bobillo, Carlos Bobed, Eduardo Mena

Dept. of Computer Science & Systems Engineering, University of Zaragoza, Spain
{ryus, fbobillo, cbobed, emena}@unizar.es

Abstract. The advantage of using semantic reasoners based on Description Logics (DL) for the development of intelligent systems is doubtless: They make easy the optimal management of knowledge (expressed as ontologies). Reasoning is a complex and computationally expensive task traditionally performed on powerful server and desktop computers. However, we should not discard reasoners from being used on mobile devices such as smartphones or tablets, as the increasing number of applications run on wireless environments demands that many intelligent tasks should be performed on mobile devices rather than on desktop or server computers. Although less powerful that their fixed counterparts, mobile devices are becoming more and more capable of running complex tasks, such as DL reasoning. In this paper we introduce the framework of a competition for reasoners on mobile devices, based on the OWL Reasoner Evaluation, with the main goal of promoting the development of reasoners adapted to mobile environments. We detail the format of such a competition including the kind of tasks that participating reasoners should execute, interesting results of such executions, and the kind of ontologies and devices that could be used to run such tests.

1 Introduction

Modern ontology languages are based on *Description Logics* (DLs) [3], a family of logics for representing structured knowledge. DLs allow the automatic discovery of implicit knowledge by using several reasoning tasks, providing a good tradeoff between expressivity and efficiency of the reasoning. However, since the theoretical complexity is usually high, it is very important to evaluate the performance of the reasoners in practice. Since 2012, the OWL Reasoner Evaluation (ORE) workshop series organizes an annual competition to evaluate the performance of DL reasoners. Such a competition tries to promote the development of new reasoners, the optimization of the currently existing ones, and the understanding of the practical benefits and drawbacks of each of them.

In the last years, there is a growing interest in the use of DL reasoners on the currently omnipresent mobile devices [1,2,15]. Using a reasoner on such devices is challenging because of their limited capabilities (compared to their fixed counterparts) and the need to adapt the reasoners. There has been some effort to port existing semantic reasoners to the Android operating system [5,14] (to our knowledge, porting to other current mobile operating systems has not

been considered yet) and to implement new reasoners specifically designed for mobile devices [8,9,11,12,13].

However, no competition has been performed yet to test DL reasoners on mobile devices. Therefore, the objective of this paper is to discuss a framework to implement an ORE-like competition on mobile devices, discussing the most relevant factors that should be taken into account to organize such a competition.

The rest of this paper is as follows. Section 2 describes which tasks will be run in the tests and which measures will be recollected for each one. Section 3 details how to perform the described tests, including the kind of ontology set and the kind of mobile devices used (we focus on Android devices due to our previous experience in the field [5,14]). Section 4 details how everything is joined into a possible competition event. Finally, some conclusions can be found in Section 5.

2 What to Test

In this section we analyse the reasoning tasks that should be considered, and the different parameters to be measured in our competition.

2.1 Reasoning Tasks

Mobile applications often use the same reasoning tasks as semantic desktop applications. However, the limited capabilities of mobile devices make the use of OWL 2 profiles particularly interesting (due to their polynomial cost reasoning algorithms). We advocate considering similar reasoning tasks as in an ORE competition of desktop reasoners, but giving priority to the most important tasks in the OWL 2 profiles. Thus, based on our previous experience in the evaluation of reasoners on mobile devices [5,14], we propose to consider the following tasks: *consistency checking* (considered in the ORE 2013 and ORE 2014), to check if there is a logical model satisfying all the axioms in the ontology, as it is a major task in OWL 2 and all its profiles; *classification* (considered in the ORE 2013 and ORE 2014), to compute the complete class hierarchy according to the subsumption relation between every named class of the ontology, as it is the most important reasoning task in OWL 2 EL; and *query answering* (considered in the ORE 2013, although there were no submissions [7]), to retrieve the individuals that satisfy a given query, as it is the most important reasoning task in OWL 2 QL and OWL 2 RL (although, as far as we know, no OWL 2 QL or OWL 2 RL reasoner has already been ported to Android devices yet).

One could also think of considering some reasoning tasks that have been specifically proposed for mobile devices, for example, MiniME implements some non-standard tasks, such as abduction, contraction, and covering [11]. However, since our objective is to compare different reasoners, we should only take into account those tasks supported by a majority of the participant reasoners. Also, the specific features of mobile devices (sensors, mobility, connectivity, etc.) and mobile scenarios (highly dynamic information) could motivate the development

of new reasoning tasks in the future. For example, tasks to test *incremental reasoning* or *stream reasoning* could help in these dynamic scenarios. However, the development of mobile applications using DL reasoners is at a too preliminary stage and we do not have yet a strong evidence that any reasoning task is particularly interesting in these scenarios. Nevertheless, this issue should be revisited in the future.

2.2 Parameters to Measure

The criteria that a reasoner has to meet to pass a test in the ORE competition [7] are: 1) process the ontology without throwing an error, 2) return a result within the allocated timeout, and 3) return the correct result (based on the majority vote). In the mobile competition, we propose to use the same criteria but adapting the third point. As reasoning on mobile devices is not so developed as reasoning on desktop computers yet, we expect some of the participating systems to be direct ports from desktop computers versions. Being mobile platforms a completely different framework, we might find that the libraries and file encodings used in desktop computers are not completely compatible or aligned with the mobile ones. For example, JFact 1.2.1 obtains incorrect results for some ontologies on Android although it obtains the correct results on a desktop computer [5]. Therefore, we advocate to perform the tests on a desktop computer first and use those results as ground-truth to perform the majority vote.

The parameters considered in the ORE competition are: 1) *Robustness*, measured in terms of number of correctly processed tasks, taking into account the previous criteria for passing a test, and 2) *CPU usage*, measured in terms of average time per correctly processed task. In addition, to measure the following parameters is very interesting for the mobile competition:

- *Memory usage*: measured in terms of MB per correctly processed task. Mobile OSs, such as Android, currently impose extra constraints on the (already limited) maximum amount of memory that apps can use. The limit varies among devices but (currently) the maximum heap size for each app is around 256 MBs of RAM only. In Android, apps share pages of memory, so the Proportional Set Size (PSS)[1] (an estimation of the consumed memory by the app) would be used. Notice that both the memory usage for the loading of the ontology and the reasoning task should be measured since the former is usually done through the OWL API and thus would not present much difference among different reasoners.
- *Power consumption*: measured in terms of mW/h per correctly processed task. Power consumption is critical in mobile devices and so, energy-efficient reasoners should be promoted in the mobile competition. The measure of power consumption on mobile devices might need of external hardware connected to the battery of the device as shown in [10].

[1] `https://developer.android.com/tools/debugging/debugging-memory.html`

As we observed in [5,14], measuring times and memory consumption on mobile devices for a given task is subject to certain variance due to the OS scheduling policies (in our case Android) on running apps. Therefore, we advocate repeating each reasoning task at least three times and averaging the results for a fair comparison. In addition, it would be useful to rank the participating systems according to the results obtained for each of these parameters to see the trade-offs the reasoners pay when they focus on one or more parameters (e.g., power consumption vs. memory usage).

3 How to Test

In this section we discuss the features of the ontologies that could be included in our dataset, and the kind of mobile devices in our competition.

3.1 Ontology Set

The ORE 2013 ontology dataset had 200 ontologies per profile [7] (i.e., OWL 2 EL, OWL 2 RL, and OWL 2 DL –i.e., ontologies not belonging to any particular OWL 2 profile–), whereas the ORE 2014 ontology dataset contains 16555 ontologies [4]. For the proposed competition, we propose to select a sample from the ORE 2014 ontology set because data about execution on desktop computers is available. In ORE competitions, every ontology had at least 100 logical axioms and 10 named concepts, and were classified according to their number of logical axioms as *small* (≤ 500), *medium* (between 500 and 4999), and *large* ontologies (≥ 5000). When evaluating mobile devices, however, it is also necessary to pay attention to the fact that the size of the file is directly related to the memory needed to load such ontology. Hence, to avoid unnecessary memory problems, it is worth to consider normalizing the ontologies by using short prefixes, the least verbose syntax, combining several axioms into an equivalent one, etc.

ORE 2012 and ORE 2013 competitions combined an offline and a live competition, but ORE 2014 was restricted to a live competition. From previous experiments on mobile devices[2], we have seen that classification and consistency checking tasks for small and medium ontologies take times within the range of 30 seconds per ontology, while the times for large ontologies are in the range of 2 mins[3]. Notice that the previous average times consider only the ontologies that could be processed in a reasonable time (less than 10 minutes). Unsurprisingly, as the size of the ontology grows, more tasks cannot be completed. Thus, such a live competition on mobile devices should be based mainly on small and medium-sized ontologies for which the reasoners are significantly faster. However, our previous experience evaluating reasoners on mobile devices [5,14] shows that the case of large ontologies is particularly interesting for several reasons. Firstly, a notable number of ontologies cannot be loaded by some reasoners or

[2] `http://sid.cps.unizar.es/AndroidSemantic`

[3] Bear in mind that these average ranges are dependent on the reasoner, the ontology (its profile and size), and the task performed.

they produce a timeout. Secondly, the variance of the results increases and so do the differences with the same reasoner on a desktop computer. Consequently, we propose to combine an offline competition, using an ontology set with about 200 small and medium ontologies (equally represented) and about 50 large ontologies, and a live competition, using an ontology set with about 50 small and medium ontologies, and 10 large ontologies. In order to avoid any bias, the ontologies for both off-line and live competition should be randomly selected from the ORE 2014 dataset according to the previously presented criteria.

Finally, new benchmarks considering the specific features of mobile devices and scenarios can be considered as well.

3.2 Mobile Devices

Several tablets and smartphones should be used in the competition, since the hardware of these two types of mobile devices is slightly different (tablets usually have more battery and memory). As new devices are released every year with improved features and the results of the competition should not be outdated until next editions, we should consider medium and high-end devices. Thus, we propose to perform the tests for, at least, one representative device of each type and device profile (i.e., at least four mobile devices with different features).

Regarding the OS, the Android operating system [6] presents several benefits: 1) its diffusion (52% of devices use this operative system according to a recent estimation[4], with a prevision of a heavy increase according to its shipment share of 78% during the first quarter of 2015[5]), 2) its openness and thorough documentation, and 3) the existence of a Java-like native virtual machine (Dalvik) that makes it easier to reuse existing Java applications, something very important since most of the semantic APIs and reasoners have been developed in this language. Regarding the specific Android version to use we advocate using at least Android 4.x as it represents around 80%[6] of current Android devices. In a future, version 5.x may be interesting due to the replacement of the Dalvik virtual machine by the new runtime environment called Android Runtime (ART) which could boost the performance of some apps according to some first reviews.

One could consider fixing a standard hardware for the competition, making it easier to optimize the reasoners for such devices. However, we prefer testing the reasoners on some of the more commonly used devices rather than indirectly suggesting users to update to devices for which the reasoners have been specifically optimized.

4 Competition

In the ORE competition (on fixed devices), competitors must submit an executable *wrapper* with a predefined behavior which can be easily invoked from

[4] `https://www.netmarketshare.com`
[5] `http://www.idc.com/prodserv/smartphone-os-market-share.jsp`
[6] `http://developer.android.com/about/dashboards/index.html`

different scripts to automate the tests. For a competition on Android mobile devices, we propose a slightly different approach. In the following we explain the main steps of the proposed competition (see Figure 1):

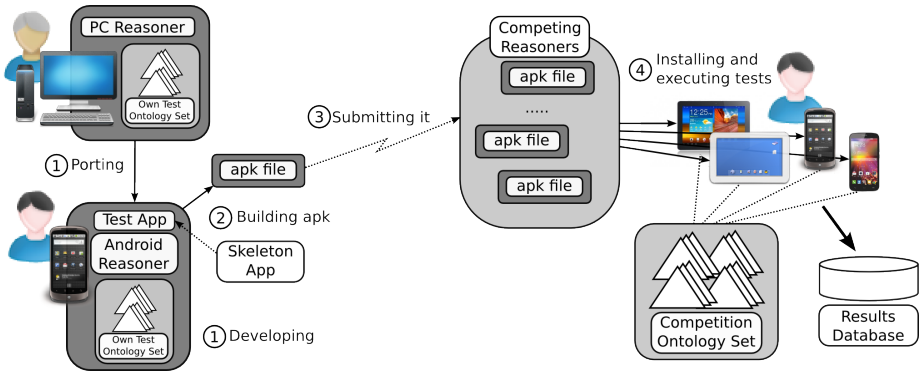

Fig. 1. Steps of the proposed competition on mobile devices.

1. The first step for competitors is to develop their systems directly for the Android platform, or to port their existing systems for desktop computers to Android. For the latter, competitors can consider our previous experience on the matter [14] and test their systems against some sample ontologies.
2. An `apk` file (a self-contained packaged application) must be built. For this task we will provide an Android app skeleton that competitors have to extend by implementing a Java interface to call their own reasoner. This `apk` can be installed and tested on the mobile devices of competitors, obtaining the input ontologies from the URIs specified in a configuration file. At this test, competitors can only use their own sample ontologies for testing.
3. Before submitting the final `apk` file, it must be configured to run against a URI (provided by the competition organizers) referencing the official competition ontology set. The set of ontologies will be empty before submission deadline to avoid optimizing the reasoners for these particular cases.
4. All the received `apk` files will be installed and invoked by referees to execute the reasoning tasks against the competition ontology set on different mobile devices. The results will be sent to a centralized database for further analysis.

This way, by providing a template skeleton, we allow competitors to easily build a full app that they can run on their own mobile devices and against their own ontology sets. Moreover, they can use any tool they need (e.g., native code implementations, libraries, etc.), as long as it is included in the `apk` file. Indeed, reasoner developers can consider using native code for the core of the reasoner submitted as it greatly improves the performance of applications on Android, although using native code also has some drawbacks such as a lower portability of the reasoner.

This competition of reasoners on mobile devices should be co-located with the classical ORE, being a possible meeting point for two different communities such as mobile computing and Semantic Web, and an important forum where to discuss the needs and constraints that mobile environments imposes on reasoning tasks. Indeed, this would help to broaden further the use and adoption of Semantic Web technologies.

5 Conclusions and Future Work

Mobile devices are becoming powerful enough to handle semantic technologies such as reasoners. So, recently, there has been a growing interest in the use of DL reasoners on mobile devices. In this paper, we have proposed an ORE-like competition for semantic reasoners on mobile devices which tries to promote: 1) the development of new reasoners specially designed for mobile devices and 2) the adaptation of traditional reasoners to this challenging scenario. We have detailed the competition explaining:

- The reasoning tasks to consider and the parameters to measure to show the consumption of the limited resources.
- The kind of ontology set and the kind of mobile devices that should be selected to obtain significant results for mobile app developers.
- A competition schema to enable reasoner developers to submit their reasoners in a easy (using an app skeleton) and flexible way (they can use any tool they need in their apps) without burdening the final testers.

Our proposal has been justified taking as basis both our previous experience in the evaluation of reasoners on mobile devices and the growing interest of different research areas, such as health-care.

In the future, we would like to crystallize these ideas by organizing the first mobile semantic reasoner competition at the ORE workshop, after discussing and adapting such a competition to the ideas and suggestions of the ORE community.

Acknowledgments. This research work has been supported by the CICYT project TIN2013-46238-C4-4-R and DGA-FSE. We also want to thank the anonymous reviewers for their comments and suggestions.

References

1. Abidi, S.R., Abidi, S.S.R., Abusharekh, A.: A Semantic Web based mobile framework for designing personalized patient self-management interventions. In: Proceedings of the 2013 Mobile and Information Technologies in Medicine and Health Conference (MobileMED 2013) (2013)
2. Ambroise, N., Boussonnie, S., Eckmann, A.: A smartphone application for chronic disease self-management. In: Proceedings of the 2013 Mobile and Information Technologies in Medicine and Health Conference (MobileMED 2013) (2013)

3. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F.: The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
4. Bail, S., Glimm, B., Jiménez-Ruiz, E., Matentzoglu, N., Parsia, B., Steigmiller, A.: Summary ORE 2014 competition. In: Proceedings of the 3rd International Workshop on OWL Reasoner Evaluation (ORE 2014). vol. 1207, pp. IV–VII. CEUR Workshop Proceedings (2013)
5. Bobed, C., Bobillo, F., Yus, R., Esteban, G., Mena, E.: Android went semantic: Time for evaluation. In: Proceedings of the 3rd International Workshop on OWL Reasoner Evaluation (ORE 2014). vol. 1207, pp. 23–29. CEUR Workshop Proceedings (2014)
6. Burnette, E.: Hello, Android: Introducing Google's Mobile Development Platform. The Pragmatic Programmers, LLC. (2010)
7. Gonçalves, R.S., Bail, S., Jiménez-Ruiz, E., Matentzoglu, N., Parsia, B., Glimm, B., Kazakov, Y.: OWL Reasoner Evaluation (ORE) workshop 2013 results: Short report. In: Proceedings of the 2nd International Workshop on OWL Reasoner Evaluation (ORE 2013). vol. 1015, pp. 1–18. CEUR Workshop Proceedings (2013)
8. Motik, B., Horrocks, I., Kim, S.M.: Delta-reasoner: A Semantic Web reasoner for an intelligent mobile platform. In: Proceedings of the 21st World Wide Web Conference (WWW 2012), Companion Volume. pp. 63–72 (2012)
9. Müller, F., Hanselmann, M., Liebig, T., Noppens, O.: A tableaux-based mobile DL reasoner - An experience report. In: Proceedings of the 19th International Workshop on Description Logics (DL 2006). CEUR Workshop Proceedings, vol. 189 (2006)
10. Patton, E., McGuinness, D.: A power consumption benchmark for reasoners on mobile devices. In: Proceedings of the 13th International Semantic Web Conference (ISWC 2014), Part I. Lecture Notes in Computer Science, vol. 8796, pp. 409–424. Springer-Verlag (2014)
11. Ruta, M., Scioscia, F., Sciascio, E.D., Gramegna, F., Loseto, G.: Mini-ME: the mini matchmaking engine. In: Proceedings of the 1st OWL Reasoner Evaluation Workshop (ORE 2012). vol. 858. CEUR Workshop Proceedings (2012)
12. Sinner, A., Kleemann, T.: KRHyper – In your pocket. In: Proceedings of the 20th International Conference on Automated Deduction (CADE-20). Lecture Notes in Computer Science, vol. 3632, pp. 452–458. Springer-Verlag (2005)
13. Steller, L., Krishnaswamy, S., Gaber, M.M.: Enabling scalable semantic reasoning for mobile services. International Journal on Semantic Web and Information Systems 5(2), 91–116 (2009)
14. Yus, R., Bobed, C., Esteban, G., Bobillo, F., Mena, E.: Android goes semantic: DL reasoners on smartphones. In: Proceedings of the 2nd International Workshop on OWL Reasoner Evaluation (ORE 2013). vol. 1015, pp. 46–52. CEUR Workshop Proceedings (2013)
15. Yus, R., Mena, E., Ilarri, S., Illarramendi, A.: SHERLOCK: Semantic management of location-based services in wireless environments. Pervasive and Mobile Computing 15, 87–99 (2014)