Situation-Aware Mobility
An Application for Stream Reasoning

Marko Luther and Sebastian Böhm

DOCOMO Communications Laboratory Europe GmbH, Munich, Germany
lastname@docomolab-euro.com

Abstract. We consider situation-aware mobile services as applications that could benefit considerably from advances in stream reasoning technology. Our initial realization of a situation-aware mobile service infrastructure combines streams of context information gained from mobile sensors and various Web resources and applies expressive reasoning. With the availability of stream reasoners, this distributed computing infrastructure could be made considerably simpler and more powerful. Recent findings in applying and evaluating existing reasoning engines are taken into account to highlight essential requirements and potential directions for future research on stream reasoning.

1 Intelligent Mobile Services

In the recent past, the mobile phone has seen a dramatic shift in its perception from a plain communication device to an intelligent source of relevant information for the very moment. Location-based services, first introduced a couple of years ago, paved the way for much more useful mobile services that assist the user in his spatial surrounding. Foreseen as the successor of these mere location centric services, context-aware applications are meant to enable a multiplicity of intelligent agents, capable to interpret heterogenous data sources and thus enabling truly intelligent services.

With IYOUIT[1], a framework for novel mobile applications, we recently introduced a mobile community service in the field of context-awareness [1]. Taking into account various types of context data, including spatial, temporal, social and environmental information, IYOUIT aims at automatically identifying the user's current situation (e.g., "sitting in a moving vehicle while commuting to work") [2]. Depending on the quality and confidence of the derived information, sophisticated new mobile services can be built and existing applications may be enriched to adapt to the user's needs. To give an example, future life-logging [3] and real-time social-networking services [4] could become even more versatile in processing complex requests like "what has been the name of that hotel we used to stay last summer together with our friends from Rome" or "is my friend already at home or still on his way".

[1] http://www.iyouit.eu

IYOUIT has been realized as a light-weight mobile client application that connects to a distributed server infrastructure [5]. Various types of sensory inputs are recorded on the mobile handset, amongst others including cell tower information, people and other devices in proximity as well as accelerometer data, to be sent to the corresponding component in the network. This data either changes periodically (eg. the accelerator sensor data is measured every minute) or event driven (eg. location updates triggered by the mobile phone connecting to another base station). Based on the aggregation, abstraction and the subsequent combination of this context information with external data sources from the Web, qualitative and highly valuable data is derived and in turn provided as input to the mobile client application to adapt to the user's current situation.

We see a high potential in the application of semantic technologies, in particular the deployment of the Web Ontology Language (OWL) [6] and expressive inference engines, to classify the wealth of collected context information. Due to the nature of this constantly changing set of information from heterogeneous data sources, the concept of stream reasoning [7] could solve a number of issues that are not addressed satisfactory by semantic technologies available today.

2 Architectural Considerations

To deploy a situation-aware service infrastructure with already existing semantic technology involves a couple of architectural design decisions, as depicted on the left-hand side of Figure 1. With mobile handsets becoming more and more powerful in terms of the number of built in sensors, the client application has been designed to gather and combine various sensor outputs. For further processing, lower-level sensor data is sent (approximately every 2 to 20 minutes) to the network infrastructure via a permanent mobile data connection. The real-time constraints for the situation computation limit the maximum classification time to not more than a second.

Due to the limitations of current reasoning engines in efficiently dealing with large data sets in expressive knowledge representations, further abstractions have to be derived before the actual reasoning can be applied. In our current architecture, the client server communication follows a standard request reply pattern, in which the mobile client triggers the actual situation detection as soon as any of the sensory inputs registers a change in its state. However, only two minimal snapshots can be compared to decide whether or not a new computation of the current situation is meaningful or not. With regard to the fact that the underlying data is often noisy, uncertain or even inconsistent, this decision is hard to make in the given setting. For example, in case no GPS fix is available, IYOUIT computes the phone's position by mapping the connected base station identifier to an estimation of its location (latitude, longitude and range) based on previous measurements that have been stored in a DB [8].

Here, stream reasoning could improve the detection of meaningful classification attempts in relocating the event driven evaluation of state changes to the networked components (cf. right-hand side of Figure 1). With far more com-

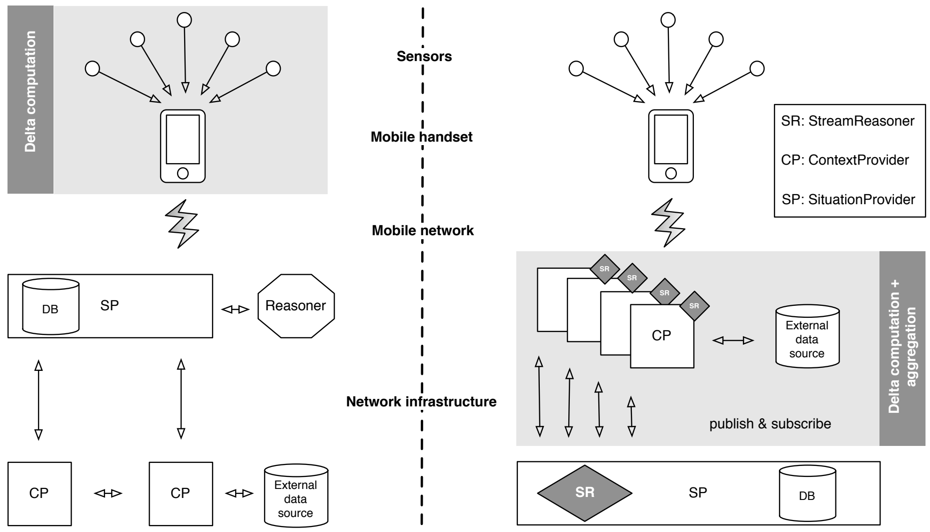

Fig. 1. Possible Architectures

puting power available and the possibility to efficiently persist data, larger timeframes could be taken into consideration. For example, the error rate of detecting a driving event based on the current speed computed by taking the delta of two location estimations is far too high, in case no GPS speed is available. If a series of geo coordinates and their estimated ranges are considered, a driving event can be detected reliably with simple geometrical calculations that assume a directed move. Furthermore, a subscription based model of interconnected components might distribute the previously centralized reasoning task.

3 Performance Considerations

To find the optimal selection of an OWL reasoner and service interface for our application that comply with the real-time constraints we conducted extensive performance evaluations with knowledge bases of different size and complexity [9]. These results confirm earlier findings [10,11] that there is still no reasoner available, capable to deal with large and complex extensional knowledge in a robust manner and that the currently available OWL reasoners still fail to answer a number of small or trivial test cases correctly. As a consequence, we designed our knowledge bases to be as small and light-weight as possible, only using language constructs that are strictly needed, to keep the inference complexity low. A drawback of this is that our knowledge bases do not link to standard ontologies to avoid an increase in complexity and therefore do not profit from a well-defined cross domain vocabulary.

It became also apparent in our study [9] that the utilized reasoner connection and the amount of data that needs to be transferred may have an enormous impact on the overall performance. One recent trend to improve this are in-memory interfaces of reasoners that clearly outperform networked connections via TCP or HTTP, resulting in a less flexible distribution of components. Another way to reduce the communication overhead is to avoid the retransmission of the same axioms over and over again. In our application it is possible to clearly separate the (small set of) changing data from the (larger body of mostly) invariable world knowledge. Therefore, we may transmit the world knowledge only once and transmit the limited amount of changing axioms separately, using axiom-retraction to clean up the reasoner state after having retrieved the inference result. Of course, both the reasoner and its communication protocol need to support axiom-level transmission and retraction. To the best of our knowledge, RacerPro [12] is currently the only OWL engine that supports retraction via all its interfaces, including its OWLAPI[2] connector as well as the standard OWLlink[3] protocol [13]. Interestingly, RacerPro communicating verbose OWL XML syntax via HTTP using OWLlink's retraction mechanism outperformed the faster Pellet reasoner [14] connected in-memory via the OWLAPI.

Most OWL reasoning algorithms have been designed and optimized for relative static knowledge bases.

For applications with frequently changing data like expressive syndication [15], incremental query answering and incremental reasoning algorithms using model-caching techniques were developed [15,16].

Todays incremental query answering algorithms are limited to theories with only simple roles whereas our examples rely on transitive rules. Here, the incremental reasoning algorithms under syntactic addition and removal of extensional knowledge implemented as part of the incremental consistency checker of newer Pellet version should be able to considerable speed up our use case. However, we could not detect any specific performance gain of this implementation. Still, in case the actual reasoning task can be computed very fast and the necessary axioms representing the changing data can be transmitted efficiently, the retrieval of large inference results back to the application accounts for a time critical task in the overall process.

4 Discussion

We introduced an application scenario that could benefit from advances in the field of stream reasoning. Especially, the use of stream reasoners following the revolutionary approach [7], communicating data directly as axioms expressed in OWL or less expressive representation languages, would considerably simplify and improve our current system architecture.

Our recent performance results clearly demonstrate the need for reasoning support in modular theories, fast standard axiom level communication protocols supporting retraction (such as OWLlink), and incremental inference algorithms as enabling technologies for stream reasoning. At the same time we want to

[2] `http://owlapi.sourceforge.net`
[3] `http://www.owllink.org`

raise our concerns regarding the correctness of even more sophisticated inference engines, especially, if inference results are used in critical ways (e.g., for context-dependent access control mechanisms).

References

1. Böhm, S., Koolwaaij, J., Luther, M., Souville, B., Wagner, M., Wibbels, M.: Introducing IYOUIT. In: Int. Semantic Web Conf. (ISWC'08), October 27–29, 2008. Volume 5318 of LNCS., Springer (2008) 804–817
2. Luther, M., Fukazawa, Y., Wagner, M., Kurakake, S.: Situational reasoning for task-oriented mobile service recommendation. The Knowledge Engineering Review **23** (2008) 7–19
3. Gemmell, J., Bell, G., Lueder, R.: MyLifeBits: a personal database for everything. Communications of the ACM **49** (2006) 88–95
4. Koolwaaij, J., Tarlano, A., Luther, M., Nurmi, P., Mrohs, B., Battestini, A., Vaidya, R.: ContextWatcher – sharing context information in everyday life. In: Proc. of the IASTED International Conference on Web Technologies, Applications, and Services (WTAS'06). (2006)
5. Böhm, S., Koolwaaij, J., Luther, M.: Share whatever you like. In: Proceedings of the First International DisCoTeC Workshop on Context-aware Adaptation Mechanisms for Pervasive and Ubiquitous Services (CAMPUS'08). (2008)
6. Patel-Schneider, P.F., Hayes, P., Horrocks, I.: OWL Web Ontology Language: Semantics and Abstract Syntax. W3C Recommendation, 10 February 2004, World Wide Web Consortium (2004) <http://www.w3.org/TR/owl-semantics/>.
7. Della Valle, E., Ceri, S., Barbieri, D., Braga, D., Campi, A.: A First step towards Stream Reasoning. In: Proceedings of the Future Internet Symposium. (2008)
8. Koolwaaij, J.: Mapping the GSM landscape. In: Proceedings of the SVG Open. (2005) <http://www.svgopen.org/2005/proceedings.html>.
9. Luther, M., Liebig, T., Böhm, S., Noppens, O.: Who the Heck is the Father of Bob? In: Proc. of the European Semantic Web Conference (ESWC'09). Volume 5554 of LNCS., Springer, Heidelberg (2009) 66–80
10. Weithöner, T., Liebig, T., Luther, M., Böhm, S., von Henke, F.W., Noppens, O.: Real-world reasoning with OWL. In: Proceedings of the 4th European Semantic Web Conference (ESWC'07), Innsbruck, Austria (2007) 296–310
11. Liebig, T.: Reasoning with OWL: System Support and Insights. Technical Report TR-2006-04, Ulm University, Germany (2006)
12. Haarslev, V., Möller, R.: Description of the RACER system and its applications. In: Proc. of the Int. Workshop on Description Logics. (2001)
13. Liebig, T., Luther, M., Noppens, O., Rodriguez, M., Calvanese, D., Wessel, M., Horridge, M., Bechhofer, S., Tsarkov, D., Sirin, E.: OWLlink: DIG for OWL 2. In: Proc. of the OWL Experiences and Directions Workshop at the ISWC'08. (2008)
14. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. Journal of Web Semantics **5** (2007) 51–53
15. Halaschek-Wiener, C., Kolovski, V.: Syndication on the Web using a Description Logic approach. Journal of Web Semantics **6** (2008) 171–190
16. Halaschek-Wiener, C., Parsia, B., Sirin, E.: Description Logic reasoning with syntactic updates. In Meersman, R., Tari, Z., eds.: OTM Conferences. Volume 4275 of LNCS., Springer (2006) 722–737