Mini-ME 2.0: powering the Semantic Web of Things

M. Ruta, F. Scioscia, G. Loseto, F. Gramegna, S. Ieva, and E. Di Sciascio

Politecnico di Bari, via E. Orabona 4, I-70125 Bari, Italy
E-mail: (michele.ruta, floriano.scioscia, giuseppe.loseto, filippo.gramegna, saverio.ieva, eugenio.disciascio)@poliba.it

Abstract. This paper presents an updated version of Mini-ME, a mobile reasoner for the Semantic Web of Things. Building upon previous stronger elements, *i.e.*, computational efficiency and support for non-standard inference services, novel features have been added. Particularly, the *Concept Covering* reasoning task for request answering via service/resource composition has been included among allowed inferences, *Protégé* plugins have been released and the support for *OWLlink* protocol is now available. As a proof of concept, two use cases are presented, both in the mobile and ubiquitous computing field: a wireless semantic sensor network and a mobile semantic augmented reality scenario.

1 Introduction

The Semantic Web of Things (SWoT) vision is joining the Semantic Web and the Internet of Things paradigms. It enables semantic-enhanced pervasive computing by embedding intelligence into ordinary objects and environments through a large number of heterogeneous micro-devices, each conveying a small amount of information. Application domains include wireless sensor and actor networks, home and building automation, mobile service/resource discovery, among others. In those scenarios, reasoning engines are not exploited only for query answering, but also as decisional and organizational systems. Hence, standard reasoning services such as *Subsumption* and *Satisfiability* checking are not enough. Furthermore, mobile and embedded devices are basically resource-constrained, so they can run properly only optimized inference engines, while common reasoners generally impose not-trivial hardware and software constraints. In order to provide advanced matchmaking and resource retrieval functions for the SWoT, in [12] *Mini-ME (the Mini Matchmaking Engine)*[1] was presented, a compact matchmaker and reasoner for $\mathcal{ALN}$ Description Logic. The system was evaluated as performing well w.r.t. widespread reasoners such as *Pellet*, *Hermit* and *Fact++* in standard reasoning tasks such as Concept Satisfisability and Subsumption test, Ontology Coherence test and Classification (full results are in [12], not replicated here). Furthermore, it supported *Concept Abduction* and *Concept Contraction* non-standard inference services, which allow it to support

---

[1] http://sisinflab.poliba.it/swottools/minime

more advanced semantic matchmaking [2] w.r.t. other reasoning engines. Mini-ME uses the OWL API [5] to parse and manipulate Knowledge Bases in OWL 2 supported syntaxes. It exploits structural inference algorithms on unfolded and CNF (Conjunctive Normal Form) normalized concept expressions for efficient computations also on resource-constrained platforms. In [17] four Semantic Web reasoners were successfully ported to the Android platform, albeit with significant rewriting or restructuring effort in some cases. Similarly, in [6] the *ELK* reasoner was optimized and evaluated on Android. Nevertheless, all those systems were designed mainly for batch jobs over large ontologies and/or expressive languages, which made mobile devices less suitable due to slower computation and smaller memory. The non-standard services of Mini-ME are more useful in SWoT scenarios, where mobile agents provide quick decision support and/or on-the-fly organization in environments intrinsically unpredictable as the mobile ones. *Mini-ME* has been now updated, including novel features (see Section 2 for details). Main improvements comprise: (i) optimization for a more efficient memory management; (ii) software re-engineering for improved maintainability; (iii) support for the *OWLlink* protocol [7]; (iv) introduction of abduction-based *Concept Covering* inference service, described in Section 2.1; (v) implementation of a pair of plug-ins for the *Protégé* ontology editor, described in Section 2.2.

Mini-ME was employed in several prototypical testbeds in the field of Semantic Web of Things. Two use cases are presented here as proof of concept: Section 3.1 reports on a wireless semantic sensor network based on CoAP (Constrained Application Protocol) [1], while Section 3.2 overviews a mobile semantic augmented reality scenario. Section 4 closes the work.

2 Improvements and novel features

In its early version Mini-ME exposed two interfaces: *OWLReasoner* and *MicroReasoner* [12], respectively for standard and non-standard inference services. Now they have been consolidated in the latter interface which provides all the services entry points. In addition, data structures have been organized in a hierarchy which allows more flexible manipulation and efficient memory usage. Support for the OWLlink [7] protocol was integrated, based on the *OWLlink API* [9]. OWLlink allows OWL 2 [16] reasoners to offer a standard HTTP/XML-based interface to applications. OWLlink support is currently limited to the core protocol, thus allowing requests for standard inferences only. Extensions for non-standard inference services are being devised and are planned for integration in the next Mini-ME version. Further novel features are described in the following subsections and in the Mini-ME web page at http://sisinflab.poliba.it/swottools/minime, where also the results of the OWL Reasoner Competition[2] will be published.

---

[2] http://vsl2014.at/meetings/ORE-competition.html

2.1 Concept covering

Many SWoT scenarios require that relatively large number of low-complexity resources are aggregated in order to satisfy an articulated request. To this aim, in addition to *Concept Abduction* and *Concept Contraction* non-standard inferences, a further reasoning task based on the solution of *Concept Covering Problem* (CCoP, formally defined in [10]) is now available. It allows to: (i) cover (*i.e.*, satisfy) features expressed in a request as much as possible, through the conjunction of one or more instances of a Knowledge Base (KB) –seen as elementary building blocks– and (ii) provide explanation of the uncovered part of the request itself. Given a concept expression R (request) and a set of instances $S = \{S_1, S_2, \ldots, S_n\}$ (available resources), where R and $S_1, S_2, \ldots, S_n$ are satisfiable in the reference ontology $\mathcal{T}$, *Concept Covering* aims to find a pair $\langle S_c, H \rangle$ where S_c includes concepts in S (partially) covering R w.r.t. $\mathcal{T}$ and H is the (possible) part of R not covered by concepts in S_c. Algorithm 1 is applied to solve CCoP. A compatibility check is performed (line 7) to verify if a resource S_i (from set S) can cover the request. Afterwards (line 8) *abduce* algorithm –described in [11]– solves a Concept Abduction Problem (CAP) to determine what is missing in the resource description, in order to completely satisfy the request. It defines also a penalty value of S_i w.r.t. H based on the *norm* of CNF expressions [11]. Finally, the resource (S_{max}) with the lowest penalty (r_{min}) –*i.e.*, the resource best covering H at each step– is selected and moved from S to S_c (lines 17-18) and the part of H covered by S_{max} is removed (line 19). The algorithm output is the set of resources best covering the request, along with the uncovered part, if present.

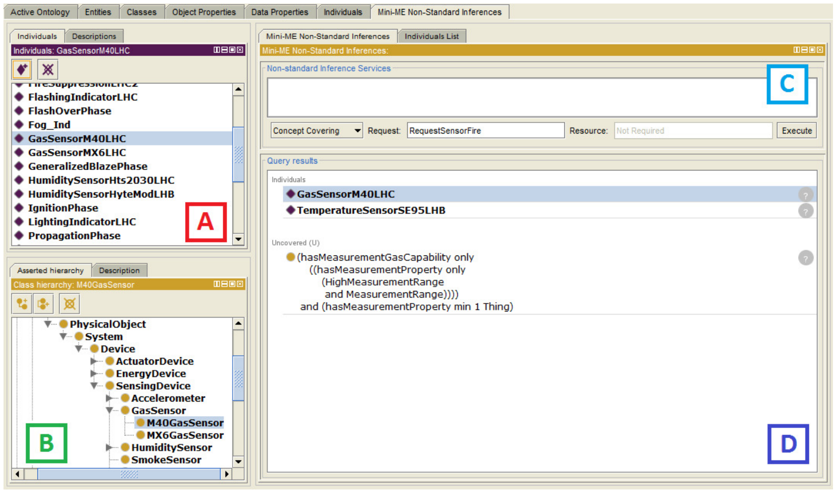
Algorithm 1 Algorithm for solving Concept Covering Problem (CCoP)

Algorithm: *solveCCoP* ($\langle \mathcal{L}, \mathcal{T}, R, S \rangle$)

```
Require:
    – $\mathcal{L}$ Description Logic;
    – acyclic TBox $\mathcal{T}$;
    – concept expression of request $R$;
    – $S = \{S_1, S_2, \ldots S_n\}$ concept expressions of available resources;
    $R$ and $S_i$ are expressed in $\mathcal{L}$ and satisfiable in $\mathcal{T}$.
Ensure:
    – $S_c = \{S_1, S_2, \ldots S_k\}$ set of resources covering the request $R$ (with $k \leq n$);
    – $H$ uncovered request.
 1: $S_c := \emptyset$
 2: $H := R$
 3: repeat
 4:     $r_{min} := norm(H, \mathcal{T})$
 5:     $S_{max} := \top$
 6:     for all $S_i \in S$ do
 7:         if $(S_i \sqcap R)$ is satisfiable in $\mathcal{T}$ and $S_i$ is a cover for $H$ then
 8:             $\langle H_i, r \rangle := abduce(\langle \mathcal{L}, \mathcal{T} H, S_i \rangle)$
 9:             if $r < r_{min}$ then
10:                 $r_{min} := r$
11:                 $S_{max} := S_i$
12:                 $H_{max} := H_i$
13:             end if
14:         end if
15:     end for
16:     if $S_{max} \neq \top$ then
17:         $S_c := S_c \cup S_{max}$
18:         $S := S \setminus \{S_{max}\}$
19:         $H := H_max$
20:     end if
21: until $S_{max} \neq \top$
22: return $S_c, H$
```

2.2 Protégé plugins

Mini-ME has been integrated within the Protégé ontology editor [4] through the implementation of an OWL reasoner plugin. It is accessible through the Protégé user interface in the *Reasoning* menu. A further Protégé plugin has been developed to exploit non-standard inferences through a user-friendly GUI (Figure 1). It was devised to support users during the development of ontology for pervasive scenarios; all supported inferences can be directly exploited and tested also through Protégé. The existing *DL Query*[3] plugin was used as guideline. The proposed plugin is a *Tab Widget* and it consists of the following components, highlighted in Figure 1: (A) *OWLIndividualsList* and *OWLIndividualsTypes* tabs, showing all KB instances with related description; (B) *OWLAssertedClassHierarchy* and *OWLClassDescription* tabs, containing the general taxonomy along with the description of selected classes; (C) an input box used to select the inference task to be executed, the request R and –in case of Concept Abduction and Concept Contraction– the resource annotation S. Both can be selected from the *OWLIndividualsList* through drag-and-drop. For Concept Covering it is instead possible to select a subset of KB individuals through the *Individuals List* panel as composing resources; (D) (results area) shows the output of the selected inference service. In Figure 1 a CCoP is solved, component individuals and the uncovered part of the request are shown.



Concept Covering
Request: RequestSensorFire
Resource: Not Required

```
Individuals
GasSensorM40LHC
TemperatureSensorSE95LHB

Uncovered (U)
(hasMeasurementGasCapability only
    ((hasMeasurementProperty only
        (HighMeasurementRange
         and MeasurementRange))))
  and (hasMeasurementProperty min 1 Thing)
```

Fig. 1: Protégé plugin for non-standard inferences

---

[3] http://protegewiki.stanford.edu/wiki/DL_Query

3 Motivating scenarios

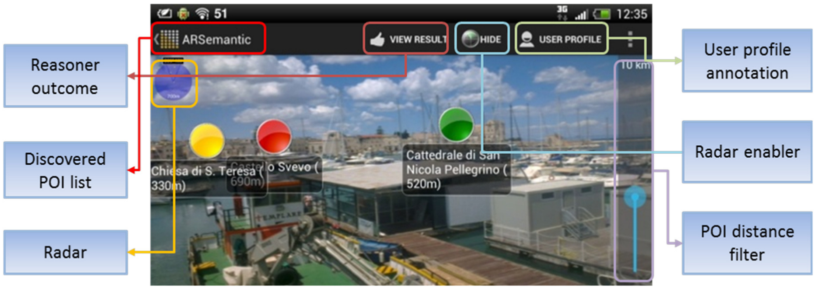
3.1 CoAP-based semantic sensor networks

The Semantic Sensor Network (SSN) paradigm [8] aims at exploiting semantics to increase flexibility and interoperability in sensor networks. A novel SSN framework was devised and proposed in [14], supporting resource discovery through semantic matchmaking. It is based on: (i) a backward-compatible extension of the HTTP-like Constrained Application Protocol (CoAP) [1] for resource discovery; (ii) non-standard inference services for retrieving and ranking resources; (iii) adoption of W3C standard SSN-XG ontology [3] to annotate data, events and device features. Each sensor is basically seen as a server exposing both sensor readings and internal information as resources toward clients, which act on behalf of end-user applications. The standard CoAP resource discovery mechanism only allows a syntactic string-matching of attributes, lacking explicit and formal characterization of the resource semantics. A protocol enhancement has been devised to support a logic-based matchmaking between a request and one or more resource descriptions, both expressed using languages grounded on Description Logics. In a car risk prevention scenario, semantic matchmaking was carried out by running Mini-ME on a testbed comprising different *Raspberry Pi* embedded boards with small computational capabilities, connected in a CoAP-based SSN. Local or remote applications act as CoAP clients and use semantic-based discovery to search for sensors or actuators, based on annotated descriptions of their features. In standard CoAP a temperature sensor would be described just with resource type `rt=temperature` and discovery would retrieve it only if request exactly corresponded. On the contrary, in semantic-enhanced CoAP resource type would be an OWL annotation w.r.t. a domain ontology and the request semantics could be matched. Unfortunately, Subsumption test returns a *yes/no* answer, so supporting only *subsume* (a.k.a. full) matches. Concept Abduction and Contraction can also identify *intersection-satisfiable* (a.k.a. potential) and *disjoint* (a.k.a. partial) matches, and also rank the resources according to the degree of similarity w.r.t. request. In the experimental evaluation, Mini-ME showed satisfactory performance in terms of processing time. Standard CoAP used on average $150ms$ to reply a basic query for temperature sensors with two resources, while $575ms$ were needed to perform a semantic CoAP discovery, executing Concept Abduction on the same –semantically annotated– resources and returning results ranked by relevance w.r.t. the request.

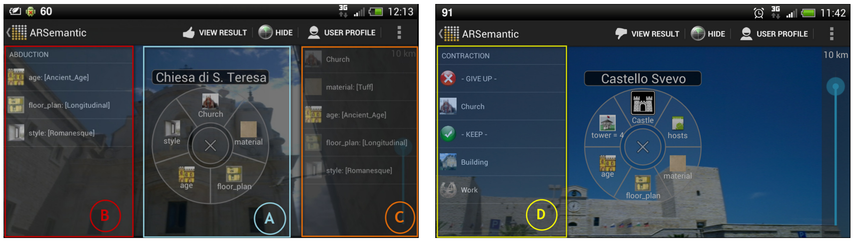
3.2 Semantic-enhanced mobile augmented reality

Semantic-based technologies can support articulated and meaningful descriptions of locations and Points of Interest (POIs). The use of metadata (annotations) endowed with formal machine-understandable meaning can enable more advanced location-based resource discovery through proper inferences. In a previous work [15], a general method and a tool were presented for annotating maps so allowing a collaborative crowd-sourced enrichment of OpenStreetMap

(OSM)[4] basic cartography. In order to allow users to exploit enriched maps, the framework is extended with a mobile Augmented Reality (AR) system for semantic-enhanced POI discovery and exploration [13]. It allows users to see an overlay of markers for POIs on the scene framed by their mobile device camera. Exploiting the embedded Mini-ME matchmaker, the mobile tool executes semantic matchmaking between the user profile and the annotations of POIs –embedded into semantic-enhanced OSM map– in her surroundings, in a reference range with respect to user's position. The user interface is shown in Figure 2a. It displays on a radar several semantic-enriched points of interest within a radius which is adjustable through a slider on the right hand side. Matchmaking outcomes are displayed as color-coded markers on the display used as device camera viewfinder, corresponding to the real direction and distance of each POI from the user. Markers for POIs within the field of sight are also shown upon the real-time device camera view. By touching a marker, the user can see its relevant features, which are presented as icons around a wheel shape, in order to provide a clear and concise description, as shown in the central portion (A) of Figure 2b. The View result panel (B) in Figure 2b lists all missing features w.r.t. user profile (C), computed through Concept Abduction. In case of incompatibility, the same left-hand menu shows Concept Contraction outcome: properties the POI satisfies and incompatible elements (Figure 2c-(D)).



(a) User interface



(b) Abduction results (c) Contraction results

Fig. 2: User interface of mobile semantic augmented reality explorer

---

[4] http://www.openstreetmap.org/

4 Conclusion and Future Work

The paper presents an improved version of the Mini-ME mobile matchmaker for the Semantic Web of Things. Added features included the Concept Covering inference, support for OWLlink protocol and Protégé plugins. Two motivating scenarios in the Semantic Web of Things field have been presented: a semantic sensor and actor networks using an enhanced version of the CoAP protocol for driving risk prevention, and a mobile semantic-enhanced augmented reality explorer. Future work includes the extension of OWLlink interface to non-standard inference services and the support for more expressive languages.

Acknowledgments

The work was supported by PON project PLATINO (PLATform for INnOvative services in future internet) and the ETCP project ARGES (pAssengeRs and loGistics information Exchange System).

References

1. Bormann, C., Castellani, A., Shelby, Z.: CoAP: An Application Protocol for Billions of Tiny Internet Nodes. IEEE Internet Computing 16(2), 62–67 (2012)
2. Colucci, S., Di Noia, T., Pinto, A., Ragone, A., Ruta, M., Tinelli, E.: A Non-Monotonic Approach to Semantic Matchmaking and Request Refinement in E-Marketplaces. Int. Jour. of Electronic Commerce 12(2), 127–154 (2007)
3. Compton, M., Barnaghi, P., Bermudez, L., Garcia-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., et al.: The SSN Ontology of the W3C Semantic Sensor Network Incubator Group. Web Semantics: Science, Services and Agents on the World Wide Web 17 (2012)
4. Gennari, J.H., Musen, M.A., Fergerson, R.W., Grosso, W.E., Crubézy, M., Eriksson, H., Noy, N.F., Tu, S.W.: The evolution of protégé: an environment for knowledge-based systems development. International Journal of Human-computer studies 58(1), 89–123 (2003)
5. Horridge, M., Bechhofer, S.: The OWL API: a Java API for working with OWL 2 ontologies. Proc. of OWL Experiences and Directions 2009 (2009)
6. Kazakov, Y., Klinov, P.: Experimenting with elk reasoner on android. In: Proc. of 2nd International Workshop on OWL Reasoner Evaluation (ORE13), Ulm (Germany). pp. 68–74 (2013)
7. Liebig, T., Luther, M., Noppens, O., Wessel, M.: Owllink. Semantic Web 2(1), 23–32 (2011)
8. Ni, L.M., Zhu, Y., Ma, J., Luo, Q., Liu, Y., Cheung, S.C., Yang, Q., Li, M., Wu, M.: Semantic Sensor Net: An Extensible Framework. International Journal of Ad Hoc and Ubiquitous Computing 4(3/4), 157–167 (Apr 2009)
9. Noppens, O., Luther, M., Liebig, T.: The OWLlink API: Teaching OWL Components a Common Protocol. In: OWLED. CEUR Workshop Proceedings, vol. 614 (2010)

10. Ragone, A., Di Noia, T., Di Sciascio, E., Donini, F.M., Colucci, S., Colasuonno, F.: Fully automated web services discovery and composition through concept covering and concept abduction. International Journal of Web Services Research (JWSR) 4(3), 85–112 (2007)
11. Ruta, M., Di Sciascio, E., Scioscia, F.: Concept abduction and contraction in semantic-based P2P environments. Web Intelligence and Agent Systems 9(3), 179–207 (2011)
12. Ruta, M., Scioscia, F., Di Sciascio, E., Gramegna, F., Loseto, G.: Mini-me: the mini matchmaking engine. In: OWL Reasoner Evaluation Workshop (ORE 2012). vol. 858, pp. 52–63 (2012)
13. Ruta, M., Scioscia, F., De Filippis, D., Ieva, S., Binetti, M., Di Sciascio, E.: A semantic-enhanced augmented reality tool for OpenStreetMap POI discovery. In: 17th Meeting of the EURO Working Group on Transportation (EWGT 2014) (jul 2014), to appear
14. Ruta, M., Scioscia, F., Pinto, A., Di Sciascio, E., Gramegna, F., Ieva, S., Loseto, G.: Resource annotation, dissemination and discovery in the Semantic Web of Things: a CoAP-based framework. In: IEEE International Conference on Internet of Things (iThings 2013). pp. 527–534. IEEE Computer Society (aug 2013)
15. Scioscia, F., Binetti, M., Ruta, M., Ieva, S., Di Sciascio, E.: A Framework and a Tool for Semantic Annotation of POIs in OpenStreetMap. In: 16th Meeting of the EURO Working Group on Transportation (EWGT 2013). Procedia - Social and Behavioral Sciences, vol. 111, pp. 1092–1101. Elsevier (feb 2014)
16. W3C OWL Working Group: OWL 2 Web Ontology Language Document Overview (Second Edition), http://www.w3.org/TR/owl2-overview/, W3C Recommendation 11 December 2012
17. Yus, R., Bobed, C., Esteban, G., Bobillo, F., Mena, E.: Android goes semantic: Dl reasoners on smartphones. In: Proc. of 2nd International Workshop on OWL Reasoner Evaluation (ORE13), Ulm (Germany). pp. 46–52 (2013)