Semantic technologies and techniques for interoperable information in smart environments

Felipe Martín Anaya
Technological Innovation Department
Indra Sistemas S.A.
Madrid, Spain
fmanaya@indra.es

Jorge Juan Rodríguez Vázquez
Technological Innovation Department
Indra Sistemas S.A.
Madrid, Spain
jjrodriguezv@indra.es

Abstract—**The current paper attempts to establish desirable features for software tools used in semantic webs. It would allow to establish benchmarks to choose the optimal solution for a specific situation, in this case the interoperability or skill to deal with different environments. The SOFIA project, having faced this situation, wants to share the chosen solution.**

Keywords-Interoperability, Ontology, Reasoner, SOFIA, Web semantics

I. INTRODUCTION

The SOFIA project is a three-year ARTEMIS project started in January 2009 and involving twenty two partners from four different EU countries. SOFIA proposes an Internet-like revolution in physical space, aiming to make “embedded information” in the physical world available for smart services – connecting the physical world with the information world and envisages that the way embedded systems are used to construct will change, due to the ongoing digital convergence, not just only physical devices but also services.

At the core of SOFIA project is the notion of smart environment. A smart environment is an ecosystem of interacting objects - e.g. sensors, devices, appliances and embedded systems in general - that have the capability to self-organize, to provide services and manipulate/publish complex data. Smart environments aspire to provide individuals with a more satisfying experience in their everyday lives from every environment (home, work), functioning invisibly and unobtrusively in the background and freeing them from hazardous work, and tedious routine tasks.

The mission of SOFIA project is to create a semantic interoperability platform and selected set of vertical applications to form an embedded system based smart environment. The key factor in these smart environments will be common, open information storage and search extent for all embedded systems, regardless of their specific implementation technology. In this vision, simple, local mash-up applications will be built on open data and devices.

The project targets encompass [1] to connect the real physical world with the information world to enrich the user experience and to enable the user to benefit from smart environments, [2] to promote innovation while maintaining value of existing legacy, [3] to create new user interaction and interface concepts and [4] to maintain cross-industry interoperability by creating an Interoperability Open Platform (IOP) as a platform for new services. IOP will foster innovation and will guarantee the future evolution of smart environments based on embedded systems, both from a scientific/technological point of view and in terms of business.

The project captures the specific aspects of smart spaces and combines the requirements for common solutions. Three applications or “verticals” are identified, which represent different kinds of space – in terms of scale, potential applications and services:

- Local personal spaces (e.g. car)
- Smart indoor spaces (e.g. home, office, university)
- Smart city (e.g. extended infrastructure and facilities like a subway station, shopping centre).

The key outcomes of the project relate to user interaction paradigms for interacting in smart environments, the common interpretability solution between many heterogeneous devices and embedded systems, and on the application development schemes that can mobilise new developers for smart environments.

II. RELATED WORK

There are several currently-available technologies that can be used to satisfy the semantic needs of the project SOFIA. Each of these uses a different technique, covering a different area in a different way. While some of them might be more efficient in time consumption, others will have a more strict consistency checking or will cover areas neglected by others, e.g. some knowledge representation languages.

This paper attempts to offer an outlook of several currently-available semantic technologies, trying to understand the differences between some of them. Such intent will never be exhaustive, as the technologies are alive, evolving as this is being written and resulting in unavoidable changes in the conclusions: We might discard one program, for instance, for not having a native implementation of RDF, but maybe next week a new release of that same software will include that feature.

Notwithstanding this, the goal can be extended to analyze the interoperability of the technologies. SOFIA is partially based in the following postulates, known as the Smart Environment Axioms:

- Interoperability is unavoidable step in the evolution of information world.
- There is no single technology that can master the variety of needs.
- World cannot be built or changed instantaneously.
- The cost of interoperability agreements increases when moving towards implementations.
- A single company cannot build the world.

This is one of the challenges that SOFIA is to face. Several technologies will be used in its environment and it will be up to SOFIA to deal with all of them, understanding all the information that they can provide and, in turn, giving them information and/or commands that they will understand. This paper will not consider exactly where is the border between the responsibilities and attributions of SOFIA, its Human users and its non-Human interlopers.

Thus, we have researched several other papers, including Dr. Raúl García Castro's thesis Benchmarking Semantic Web technology (2008) and Prof. Stefan Poslad's Ubiquitous Computing: Smart Devices, Environments and Interactions (2009), among others. Each of those works has partially covered of the interests that SOFIA has in semantic technologies. Several official websites have also been consulted, including websites of products and of the World Wide Web Consortium. Please refer to the bibliography for a complete list of consulted works.

III. TECHNICAL APPROACH

A semantic technology, as considered in this environment, will have to obtain information coded in a knowledge representation language. There are many of these, with important differences: Some of them are based only on frames, e.g., and other use also description logics.

Some of the most important ontology languages may be RDF [2] (Resource Description Framework), designed as a metadata data model; its extension RDFS (RDF Schema), that can describe ontologies; OWL [3] (Web Ontology Language), the W3C de facto standard for life sciences; its several extensions OWL DL, OWL Full, OWL Lite and OWL 2 [4]; XML Schema, etc.

Other tools will use representation models beyond those specifically designed for this subject, including Open Biomedical Ontologies (OBO) or Unified Modeling Language (UML), which only add to the difficulties of SOFIA.

One of the greatest problems when trying to apply a common layer to several ontologies comes when the current technology cannot change from any ontology to another, be it because of some imperfection in the software or because the application had not been designed to support one of the languages. Some technologies do support extended languages, but not natively, which reduces the efficiency: While translation is not needed in this case, the tool has execution problems with components that it cannot model [5].

IV. EVALUATION

An objective comparison would have to define the best practices and unambiguously test them, either in a numeric performance (for instance, measuring response time to a set input) or in a polar way (for instance, the yes-no question "Does X support the language Y?"). This will allow to offer a common benchmark, but even then it will not serve unambiguously to satisfy the needs of any specific project (i.e. SOFIA), as each project would be interested in different, sometimes weighed, features.

García suggests that two of the most important benchmarks should be the ability to import and export a given language, in his case RDF. A tool might support RDF either natively (such as Jena [6] or Sesame [7]) or non-natively (such as Protégé or WebODE), and this difference is not at all trivial in this case as has been explained before [8].

Another essential benchmark suite, supported by both García and Poslad [9], is the interoperability. This can be described as the ability of two systems to exchange, and use, information [10] or, in our field, the ability of two autonomously-developed software components to do so [11]. Interoperability, being clearly a need to exchange ontologies, is also a problem of the semantic web [12].

The cooperation of different components to reach a common goal must be achieved through a coordination of interactions using explicit communications, either through the use of a central coordinator (orchestration) or a distributed one (choreography). Unfortunately, there is not even a de facto standardization to specify which the best solution is [13].

There are several factors affecting interoperability, being first and foremost the heterogeneity of the formalisms to models the system information [14]. This heterogeneity can be found not only in the information level, but in the system level too [15].

Another factor also affecting interoperability goes beyond the sign to reach the meaning of it: More than occasionally, an ontology will not have an exact equivalent for a meaning purported by a different ontology, which could lead to an important problem[16].

The architecture M3, used by SOFIA, is the implementation of the architecture OIP developed by Nokia and VTT. SOFIA follows the M3 principles that deal with interoperability in several points, specifically: [17]

- Interoperability agreements on information level:
    - Common ontology model and data presentation format are the only requirements
    - Respecting the integrity and independence of devices

- Enabling cross-domain use cases:
    - Means and techniques must be use case independent
    - Support for enforcement to device and smart object manufacturers

Several attempts have been made to unambiguously compare the existing technologies and techniques, but these might be already obsolete due to the further changes on the software and, even if they are not, might not adjust to the specific needs of the current project. As such, while the current paper will still analyze the results of previously-established benchmark suites, it will use its own methodology to measure the points that might be of interest for SOFIA In your paper title, if the words “that uses” can accurately replace the word “using”, capitalize the “u”; if not, keep using lower-cased.

V. CONCLUSIONS

García [18] analyzed several semantic technologies in terms of importing and exporting languages, and centering in their approach to RDF. This feature favored the frameworks supporting native RDF, such as the open-sourced Jena and Sesame. The results also analyzed several other semantic web tools that non-natively supported RDF, including KAON [19] (which has been superseded by KAON2 since the publication of this work) [20]; Protégé [21] (the most extended editor); and WebODE.

About imports, all the three failed when importing class hierarchies with cycles; Protégé also failed when importing instances of multiple classes; and WebODE had several problems with properties lacking domain and range. Both KAON and Protégé were unable to export classes or instances of multiple domains; Protégé also failed with instances of multiple classes; and KAON had problems with multiple domains, undefined range or XML Schemes. WebODE, the less extended of the trio, had no problems when exporting.

Protégé was the only program that showed trouble in integration, particularly with a class which was an instance of multiple metaclasses; with data type properties without domain but with range; with data type properties whose range was String; and with instances related to those. This approach was made when trying to integrate Protégé with KAON.

When facing this problem, SOFIA approaches the question analyzing the different reasoners:

- Bossam Reasoner, a RETE-based rule engine using native supports to reason over OWL and SWRL ontologies, and RuleML rules. The analysis was incomplete. [22]
- FACT++, a reasoner implemented in C++ and supporting OWL-DL and theoretically OWL2 in the next release. [23]
- Hoolet Reasoner, a reasoner that can deal with OWL-DL ontologies only after translating them. [24]
- Jena, an open-source semantic web framework that supporting OWL. [25]
- KAON2 Reasoner, whose infrastructure manages OWL-DL, SWRL and F-Logic ontologies. [26]
- Pellet Reasoner, an open-source OWL-DL reasoner. [27]
- Racer Pro Reasoner, a closed-source semantic web reasoning system and information repository. [28]
- SweetRules, an integrated set of tools for semantic web rules and ontologies. [29]

TABLE I.

	Bossam	FACT++	Hoolet	Jena	KAON2	Pellet	Racer Pro	Sweet Rules
Expressivity	?	SROIQ(D)	?	Incomplete DL support	SHIQ(D)	SROIQ(D)	SHIQ	Not Applicable
OWL-DL Entailment	Partial / under development	Yes	Yes	Incomplete OWL-DL model	Yes	Yes	Yes	No
Consistency checking	?	?	Yes	Incomplete OWL-DL model	?	Yes	Yes	No
DIG Support	No	Yes	No	Yes	Yes	Yes	Yes	No
SPARQL Support	No	No	No	Yes	Yes	Yes	No	No
Rule Support	Yes (SWRL, own)	No	Yes (SWRL)	Yes (Own)	Yes (SWRL)	Yes (SWRL, others)	Yes (SWRL)	Yes (SWRL, others)
Documentation available	Yes	Not readily	Not readily	Yes	Not readily	Yes	Yes	Yes
Licensing	Free/ closed-source	Free/ open-source	Free/ open-source	Free/ open-source	Free/Closed Source	Several	Non-Free/ closed-source	Free/ open-source

a. SOFIA's Benchmark suite of reasoners.

SOFIA initially attempted to determine which of the available technology was more suitable to its specific needs, considering that the chosen solution could need some adaptation to solve any important deficiency. We soon discovered that the defining features of a semantic framework depend on the situation: There is not a single, categorically best solution, not even with adaptation, and most are lacking in terms of interoperability.

As a consequence of this, the approach has changed. Now, one of the mid-term goals of SOFIA is the development of a semantic environment allowing a high degree of interoperability – because SOFIA, by its nature, will deal with a variety of technologies using a variety of ontologies. Having identified some of the important features that can define a good semantic framework, SOFIA is currently working in the adaptation and creation of this new technology.

E.g.: SOFIA uses three autonomous ontologies for each of its "verticals" (ie. Local personal spaces, smart indoor spaces and smart city); all of these are managed using Protégé. Each of the ontologies extends from a fourth one, called core ontology, which includes the common classes. This only adds to the interoperability of the system, as does the use of the code generator.

REFERENCES

[1] Kindberg, T. & Fox, A. System software for ubiquitous computing. IEEE Pervasive Computing (2002)

[2] http://ww.w3.org/RDF

[3] http://www.w3.org/TR/owl-features/

[4] Cuenca, B. et al. OWL2: The Next Step for OWL. Journal of Web Semantics, 2008.

[5] Corcho, O. A layered declarative approach to ontology translation with knowledge preservation. Frontiers in Artificial Intelligence and its Applications, IOS Press (2005).

[6] http://jena.sourceforge.net/

[7] http://www.openrdf.org/doc/sesame/users/ch01.html

[8] García Castro, R. Benchmarking Semantic Web technology. Universidad Politécnica de Madrid. (2008)

[9] Poslad, Stefan. Ubiquitous Computing: Smart Devices, Environments and Interactions. Wiley. 2009.

[10] Several authors. IEEE Standard Glossary of Software Engineering terminology. IEEE, 1991

[11] Duval, E. Learning technology standardization: making sense of it all. International Journal on Computer Science and Information Systems. 2004

[12] Brachmann: R. Brachmann and H. Levesque. Readings in Knowledge Representation. Morgan Kaufmann, San Mateo, 1985.

[13] Poslad, Stefan. Ubiquitous Computing: Smart Devices, Environments and Interactions. Wiley. 2009.

[14] Brachmann: R. Brachmann and H. Levesque. Readings in Knowledge Representation. Morgan Kaufmann, San Mateo, 1985.

[15] Sheth, A. Interoperating Geographic Information Systems. Kluwer, 1998.

[16] Euzenat ea: J. Euzenat, T. Le Bach, J. Barrasa, P. Bouquet, J. De Bo, R. Dieng-Kuntz, M. Ehrig, M. Hauswirth, M. Jarrar, R. Lara, D. Maynard, A. Napoli, G. Stamou, H. Stuckenschmidt, P. Shvaiko, S. Tessaris, S. Van Acker and Ilya Zaihrayeu. D2.2.2 State of the art on ontology alignment. Technical report, Knowledge Web, 2004.

[17] SOFIA Technical Annex (Dec 2008), Grant agreement no.: 100017

[18] García Castro, R. Benchmarking Semantic Web technology. Universidad Politécnica de Madrid. (2008)

[19] http://www.sourceforge.net/projects/kaon

[20] http://kaon2.semanticweb.org/

[21] http://protege.stanford.edu/overview/protege-owl.html

[22] http://bossam.wordpress.com/.

[23] http://code.google.com/p/factplusplus/

[24] http://owl.man.ac.uk/hoolet/.

[25] http://openjena.org/

[26] http://kaon.semanticweb.org/

[27] http://pellet.owldl.com/

[28] http://www.racer-systems.com/products/racerpro/index.phtml

[29] http://sweetrules.projects.semwebcentral.org/