Semantic as a Service

August 29, 2015

OWNERS

HDE Labs
Place Sophie Laffitte
06560 Valbonne - France
Tel: +33 - (0) 6 43 16 94 79
ahassan@hdelabs.fr
www.hdelabs.fr

Contents

1 Biography

The founder of HDE Labs, Ali Hassan, is an experienced software engineer who has Master degree from USA in the domain of high performance computing in 2002 and a Ph.D. degree from Telecom Bretagne in France in semantic engineering in 2012. During these years, Dr. Hassan worked in leading companies and research centers in the research and development of edge technologies.

Dr. Hassan provided consulting in the area of semantic engineering for several companies in Sophia Antipolis and in France in general. This include: feasibility study for semantic based projects, using semantic technology, choosing the right reasoning system, customizing reasoning systems, and building semantic based applications. These projects are either company projects (to produce industry products) or government funded projects, such as European Union Projects Horizon 2020.

In 2015, Dr. Hassan started his own project HDE Labs (please see www.hdelabs.fr) to provide semantic technologies. The main focus of HDE Labs is to provide scalable and high performance reasoning capability for big projects such as smart city, context aware services, etc. Please see performance measures presented on the website.

2 Innovation - Semantic as a Service

We identified two major difficulties that face incorporation of semantic technology in traditional software development. These two difficulties are:

1. Semantic components such as ontology development, reasoning, and semantic query are complex and require long learning phase from normal software engineer who did not have previous training in the domain.
2. Scalability and performance doubts. Some projects suspect that existing technologies in semantic engineering might fail for large scale applications. Example projects could be smart city systems that include several countries.

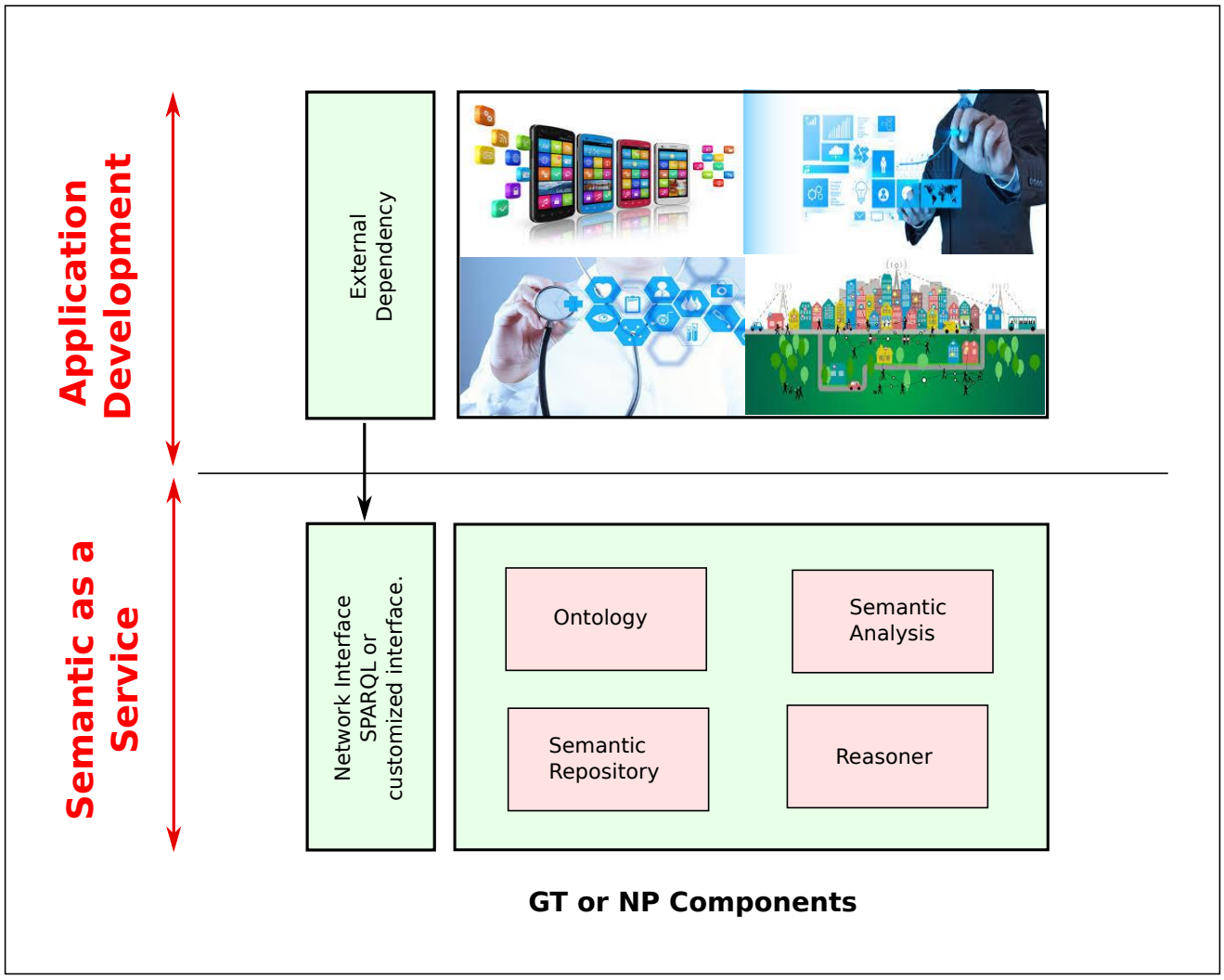

Figure 1: Semantic as a service. The complexity of semantic engineering is encapsulated inside GT and NP Components and hidden behind a very simple network interface. The application development is traditional because majority of semantic related aspects are encapsulated in NP or GT Components that are external and already accomplished.

For the first concern, the innovation in GT and NP components is that we elegantly hide all of semantic complexities from our customer. We call this approach: **semantic as a service.** All semantic related details are hidden inside GT components (or NP components). The visible part

is the service point that expose customized service that match the application needs. As result, what the customer gets is an easy to use service that fit her technological needs. Using semantic as a service approach, the customer can develop his semantic based application using traditional software development without involving in the complex details of semantic technology such as decidability of reasoning algorithms, etc. This is very important to allow majority of software companies to adopt semantic technology. Please see figure 1.

For the second concern, our strategy is to have advanced performance optimization intervention at all levels including software, hardware, and files. Also, we have developed advanced fine tuning between software and hardware. This is the result of five years of state of art R&D. The result is high performance reasoning and query semantic system. Please see tables 1 - 9. Again, semantic as a service approach is a key in this regard. All hardware and software complexities are hidden behind the easy to use interface (service).

The two factors that summarize the innovation of GT and NP components are:

1. Performance of our software. Please see tables 1 - 9.
2. Ease of use through customized interface (service). Please see figure 1.

3 Product Description

The implementation of our vision, semantic as a service, is achieved via GT and NP products. GT and NP components aims at providing semantic backend support layer for all semantic-based applications such as: smart city, mobile applications, context-aware applications, semantic interoperability, semantic search, etc. The semantic support include:

1. High performance semantic back-end support.
2. ABox and TBox hosting with IP access point.
3. Ontology development service.

It is possible to choose between:

- Normal Semantic (RDF Store). This is the NP series.
- Rich Semantic (Formal Ontology). This is the GT series.

	>1M Tuple	**>6M Tuple**
Query per Second	318.58	87.72
Query per Hour	1,156,902	315,789

Table 1: NP series optimized for query.

	>1M Tuple	**>6M Tuple**
Query per Second	10,194	2,807
Query per Hour	34,407,060	9,472,750

Table 2: NP series optimized for query. Hardware and software optimization.

	>1M Tuple	>6M Tuple
Query per Second	5.02	3.6
Query per Hour	18,094	12,972

Table 3: Best industry benchmark (based on our best knowledge).

	>1M Tuple	>6M Tuple
Query per Second	9.9	1.43
Query per Hour	35,643	5,157.59

Table 4: Best academic benchmark (based on our best knowledge).

	GT Series	Best Industry Benchmark	Best Academic Benchmark
Query per Second	0.086	x	x
Query per Hour	309.6	x	x

Table 5: GT series optimized for query. No hardware nor software optimization. Queries are performed on average Benchmark Dataset. x means the benchmark does not report results for this category.

	GT Series	Best Industry Benchmark	Best Academic Benchmark
Query per Second	0.49	x	x
Query per Hour	1,764	x	x

Table 6: GT series optimized for query. Memory optimization. Queries are performed on average Benchmark Dataset. x means the benchmark does not report results for this category.

	GT Series	Best Industry Benchmark	Best Academic Benchmark
Query per Second	1.3	x	x
Query per Hour	4,680	x	x

Table 7: GT series optimized for query. Memory and software optimization. Queries are performed on average Benchmark Dataset. x means the benchmark does not report results for this category.

	GT Series	Best Industry Benchmark	Best Academic Benchmark
Query per Second	36.648	x	x
Query per Hour	131,760	x	x

Table 8: GT series optimized for query. Memory, software, and hardware optimization. Queries are performed on average Benchmark Dataset. x means the benchmark does not report results for this category.

	GT Series	Best Industry Benchmark	Best Academic Benchmark
Query per Second	1,162.48	x	x
Query per Hour	4,200,975	x	x

Table 9: GT series optimized for query. Advanced mix of optimizations. Queries are performed on average Benchmark Dataset. x means the benchmark does not report results for this category.