pinWeb: Knowledge Fusion Platform for *H&M*

Mandeep Singh, singh@informatik.uni-tuebingen.de

This paper describes *pinWeb*, which is being developed at the *University of Tübingen, Germany*. Our goal is to create a knowledge fusion platform, which can be utilized by humans and machines (*H&M*) to manage, share and merge knowledge. By means of various existing tools and applications both *H&M* can collect knowledge in diverse domains, which must not necessarily be similar in its representation. The issue of heterogeneity in knowledge representation can be solved with an extraction, translation and query approach, if the target is only to share knowledge. For instance Gruber says in *"A Translation Approach to Portable Ontology Specification"* that the problem of knowledge representation in different languages can be addressed by following a translation approach. According to Berners-Lee's vision, web content needs to be composed in a syntactically and semantically correct way, so that it is understandable by machines. By contrast, autonomous interpretation (without human intervention) of a web page proves yet to be a challenge for machines. In fact, machines need human knowledge to work in the virtual and real world, which is still human oriented. If the machines had the ability to extract ontology from some information resource using preexisting ontologies developed by humans and to translate knowledge, just as humans do while reading books and then answering queries on newly enriched knowledge, both could share and enhance their collected knowledge. Following this idea of collective intelligence of *H&M*, this paper briefly introduces an extensible plug-in *Java* based application that abstracts over different internal knowledge representations, facilitates knowledge interoperability and supports diverse reasoning methods and language independent knowledge queries with the help of existing technologies.

The key features of *pinWeb* are knowledge enrichment and sharing. Nevertheless, there are some fundamental questions, the first one is: "*How to represent and share knowledge?*". To logically organize knowledge all data in *pinWeb* is stored as an information graph and thereby we apply a Probabilistic Ontology Model (*POM*) approach to achieve abstraction. The information graph is authored in *W3C*'s Ontology Web Language (*OWL*). If knowledge is expressed in a declarative machine language as an ontology, it can be understood and shared by machines. In spite of different ontologies, all communicating parties can exchange their knowledge, if they follow up an argumentation and dialog approach to enhance an ontology or transform it to another language at runtime. In the same manner humans can also exchange and enhance their vocabulary and knowledge of a certain domain while conversing with other humans or machines. The second question is: "*How does ontology come to existence?*". Generally, to come up with an ontology or to enhance an existing one, both (*H&M*) use the following methods: *(i) Ontology Authoring* means developing ontologies by themselves *(ii) Ontology Reuse* means import or merge an existing ontology *(iii)*

Ontology Mining means extracting an ontology out of some kind of information resource (knowledge acquisition is mostly done by reading). *(iv) Ontology Refinement* means deducing an ontology out of recently authored ontologies with the help of approved knowledge. While *pinWeb*-actors (*H&M*) practice the methods listed above, the *POM* is populated with ontologies. The third question is: "*How are ontologies fusioned?*". At the current stage of development, ontologies stored in *pinWeb* are categorized as: *(i)* public ontology, readable by all actors *(ii)* private ontology, only readable by owner *(iii)* group ontology readable for certain groups *(iv)* world ontology (fusioned ontologies), merged by applying alignment algorithms and accessible by *H&M*. In this way both can simultaneously access the world ontology and refine their own ones. This is how the fusioned knowledge base is created. The final question is: "*Which language to use for ontology representation?*". Some of the features, which *pinWeb* should have, are already developed in various *OWL* based tools and applications e.g. reasoning on knowledge, extraction of ontology form text, transformation of *OWL* from and to most of the other declarative languages. These tools and applications can be integrated as plug-ins due to the fact that we use OWL as a base to represent knowledge. Reasoning and deduction on *OWL* can be achieved with many tools such as *JENA* and *RACER*. Ontology authors may at first develop and manage ontologies in external toolkits such as *Protege* and *NeOn* and ultimately import ontologies as *OWL*s into *pinWeb*. Once knowledge is transformed into *OWL*, queries can be placed in various query languages like, *SPARQL*, *SQL* and *ARQ*. Even rule based programs can profit from the knowledge composed of *OWL*s, as tools like *JESS* and *SWRL* can create rules while accessing the fusioned knowledge base. Since *OWL* is being actively developed, *pinWeb* will become stronger as the declarative power of *OWL* grows and more *OWL* based tools come into existence.

The implementation model of *pinWeb* is a plug-in architecture based on *Eclipse*. Figure 1 depicts the main components of *pinWeb*'s architecture. *H&M* can interact with *pinWeb* via *Actor Interface* in order to author, share, organize, query and explore the knowledge base (*KB*). The *Data Access Control* will control the information access of agents and a *JADE* instance will let them communicate via the Agent Communication Language (*ACL*). We use *JENA* to grantee the persistence of the *KB*. The *Abstract Knowledge eXtraction Interface* supports the ontology extraction out of some data source. The *Abstract Knowledge Translation Interface* is designed to import or export ontologies in different declarative languages.

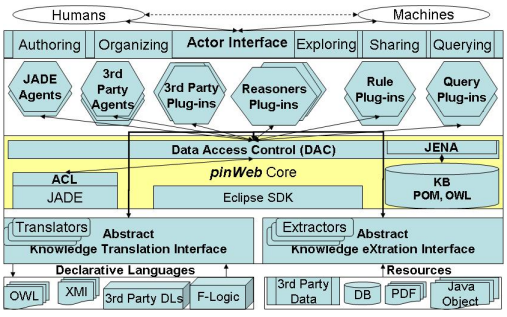

Figure 1: *pinWeb* Architecture

This paper introduces *pinWeb* that will help *H&M* to exchange knowledge in a social and complementary way. Summarizing, *pinWeb*'s strengths are its *POM* for representation, extensible architecture and support for interoperability of knowledge in *OWL*, one can clearly conclude that the potential of *pinWeb* will increase side by side to *OWL* and *H&M* will broaden each other's horizon.