Peer-to-Peer Infrastructure Supporting the Semantic Web

Johannes Ernst
R-Objects Inc.

1 CONTEXT AND MOTIVATION

Peer-to-peer technologies have recently garnered a lot of attention. Currently, most broadly adopted peer-to-peer initiatives focus on file sharing (Napster, Gnutella, many others), distributed computation (Seti@Home, Porivo, etc.), or collaboration (Groove Networks, Aimster, etc.). However, it has been argued [2][3] that the killer benefit of peer-to-peer computing will be the ability of individual users to gain back local control from centrally maintained websites, and innovate locally in spite of taking advantage of the network.

On the other side, decentralized control and innovation is what the Semantic Web needs to realize much of its potential. A "walled garden" – which is the best way to describe many web sites today – will not be able to provide the types of interoperability benefits that visionaries such as Berners-Lee foresee if/when the Semantic Web gets broadly adopted [1].

In fact, the need for decentralization is even more important for the Semantic Web than for Today's Web: not only do content authors need the ability to link from their content to another site's content, as they do on the web, but in addition:

- it needs to be easy for innovative projects/companies to link new "semantics definitions" to existing content on the web, even if the authors of the content and the meta-information are not the same; and it needs to be easy to develop software agents that use the new meta-information with existing information. If this wasn't possible, lacking a business case, the vast majority of Today's Web would remain without semantics for a long time; further, where information was indeed published with semantics, those "semantics definitions" would be essentially frozen, disallowing a lot of innovation, such as the ability by a third party to discover and represent interesting new relationships between two existing pieces of information. It is essential that this can be done without additional work by the data publisher, otherwise it won't be done in practice (imagine what it would mean for Yahoo to change their HTML code every time someone creates an addition/change to an ontology). This is a non-trivial problem as it requires the solution of a "reverse pointer traversal" problem.
- as a user, I need to be able to access semantic information published by several independent content providers, and create new semantic relationships among it for my personal use. Further, I need to be able to easily publish those relationships to people, and to software agents by no more than the push of one button. This is generally not feasible on Today's Web, and will not be easier for the Semantic Web.

Over the past several years, we have developed a pure-play peer-to-peer infrastructure that addresses these issues in a fairly novel way, which we will describe in brief in the following. Our goal with this position paper is to solicit feedback on the suitability of this approach compared to other approaches to the Semantic Web as well as the pros and cons of different approaches to integration, prior to us releasing our source code and developer's kit to the public.

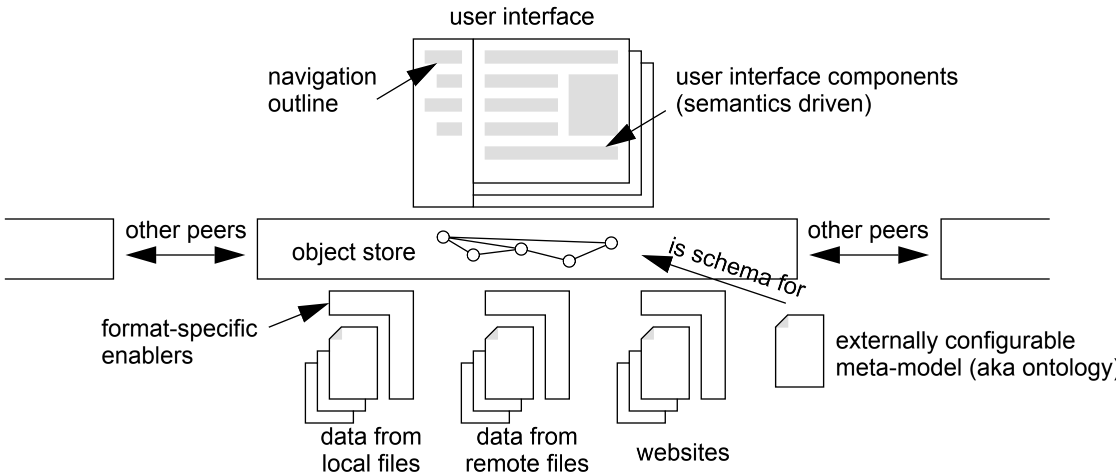
2 CORE FEATURES AND IMPLEMENTATION OVERVIEW

The R-Objects system is a pure peer system. The user interacts with a peer, implemented in Java, that runs locally on his computer. A schematic overview is shown below.

User Interface. The peer application has a component-based, fully-functional user interface which is driven by the underlying meta-model (our term for "ontology" or "semantics definition"). At run-time, the available user interface components "announce" which types of semantic objects they can interact with by identifying concepts from the meta-model. As the user accesses information, the user interface constructs itself dynamically depending on the semantics of the accessed information. This means a high degree of appropriateness of the user interface for the job at hand. It also means that the user as well as third parties can easily extend the application, while maintaining the look-and-feel and without conflicting with other parties' concurrent extensions.

Communications. A peer communicates with other peers through a partly synchronous, partly asynchronous protocol that is currently implemented on top of CORBA. This protocol allows several users to share the same object (edits are subject to distributed locking), but it also allows any user individually to relate any local object to other, local or remote objects, subject to the rules set by the underlying meta-model. This protocol also allows the identification of any object in the R-Objects network by a URI. It also provides smart replication of distributed semantic objects, once the user traverses (his, or someone else's) relationships from his local objects to remote objects. In addition to performance benefits, smart replication allows offline use of the software, which was one of our design goals.

Meta-models. R-Objects supports arbitrary meta-models, which are defined using a simple entity-relationship-attribute modeling language. This allows the use of standard information modeling tools (e.g. UML or ER tools). Any user can locally define his desired meta-model. However, a locally defined meta-model can propagate to other users when another user makes use of one of its concepts.



URLs and Today's Web. A user in the R-Objects network has access all data on the web simply by entering the appropriate URL. As this is a peer-to-peer system, the user also has access to all data on his local computer. Unlike the web itself, files from the web are represented as a web of instance of his local meta-model. There are many benefits to this, one of which is it allows the user to relate arbitrary objects in arbitrary files (not just files, or only objects in XML files). The translation between Web and R-Objects is performed in software components we call enablers, which are uni or bidirectional software converters that can be easily built and inserted.

User-specific information. In addition to accessing information from the (Semantic or Today's) Web, the user can also create new objects simply by instantiating his meta-model locally. For example, the user can instantiate arbitrary meta-relationships between (local or remote) objects, as long as the semantics of the meta-model permit a certain meta-relationship between a certain pair of objects. In particular, the user can create a relationship between objects that have been published by different authors (e.g. different websites, or between one website and an object that the user created locally). Within the R-Objects network, these relationships are always bidirectional, their integrity is guaranteed, and, crucially, their creation does not require the cooperation of the original content author(s).

Active Information and Agents. R-Objects supports both "passive information" and "active information". An object becomes active if its attributes, or its relationships to other objects depend on the attributes or relationships of one or more other objects. For example, an AlgebraicExpression meta-entity is active, and its Value meta-attribute represents the current result of evaluating the expression with respect to its arguments that are related to the AlgebraicExpression. For a user of the infrastructure, there is no difference between active and passive information, allowing the creation of complex structures of information dependencies not all that different from what Gelernter foresaw in [4], except that it driven by an explicit definition of the semantics of the information. In particular, this allows a user to create "agents" (i.e. "active objects") which derive semantic information from raw data. A simple example for that would be a meta-entity that grabs dollar numbers from HTML pages. This provides a migration path from Today's to the Semantic Web.

3 LOOKING FORWARD

R-Objects can access and use all data that is on the web today. An extension to the R-Objects systems to access 3rd-party semantics definition, such as RDF, DAML etc. (by translating it into its own internal representation) seems straightforward, although we have not implemented this at this point, and we would like to explore these opportunities further with participants of the workshop.

Most importantly, the R-Objects system has been built with decentralization in mind not only for users, but also for projects to extend the technology independently without running into central bottlenecks (such as the R-Objects development organization). This is reflected in the user interface (independent implementation of user interface components supported), the meta-model (independent extensions supported), and the types of data and meta-data that can be accessed through enablers (independent extensions supported) and our planned, upcoming release of our Java source code.

We hope to be able to work with other Semantic Web pioneers and technologies to gain critical mass for the Semantic Web to solve these challenging problems together going forward.

4 REFERENCES

[1] T. Berners-Lee, J. Hendler, O. Lassila: The Semantic Web. Scientific American May 2001.

[2] J. Ernst: Is Peer Computing Real? Invited Talk, SCI 2001, Florida, July 2001.

[3] Gartner Consulting: The Emergence of Distributed Content Management and Peer-to-Peer Content Networks. Engagement #010022501, January 2001

[4] D. Gelernter: Mirror Worlds. Oxford Uni Press, 1992.

5 ABOUT THE AUTHOR

Johannes Ernst has been excited about well-defined semantics since his time in CDIF ('93-97) that defined an integrated meta-model for complex engineering information. He discovered the natural fit with peer computing in 1998, when he started developing this described technology. He was recognized as a Technology Pioneer by the World Economic Forum in Davos in January 2001.