Common features of killer apps: A comparison with Protégé

Conference or Workshop Item

Common Features of Killer Apps: A Comparison with Protégé

Harith Alani, Kieron O'Hara, Nigel Shadbolt

Intelligence, Agents, Multimedia Group
School of Electronics and Computer Science
University of Southampton, UK
ha@ecs.soton.ac.uk

Abstract. Killer apps are highly transformative technologies that create new markets and widespread patterns of behaviour. IT generally, and the Web in particular, has benefited from killer apps creating new networks of users. The Semantic Web community on the other hand, is still unsure whether any of their applications could become a killer app. This paper sheds some light on the main aspects of killer apps in general, and compares them with the features of Protégé as a killer app for ontology curation and management.

1 Introduction

For a technology to develop a large user base, it is helpful to exist applications in which (a) the technology is essential, and (b) many users can find profit or utility. Such an application is often called a 'killer app'. Killer apps appear in the intersection between technology, society and business. In the first place, killer apps are technological, that is, artificial solutions to perceived problems or opportunities. They need not be hi-tech, they could be simple gadgets or even non-physical systems (such as double-entry book-keeping). The effect of a killer app in a domain is to increase network effects so that a large community of users develops and reaches critical mass. This community will have enough members to attract new users, not by the underlying technologies, but by the opportunities (whether commercial, scientific or just fashionable) created by the existence of the community itself.

The development of the Web was greatly boosted by a number of killer apps that increased the usability, desirability and utility of the Web, thereby attracting users beyond the academic community. Applications such as eBay allowed new forms of behaviour that users find attractive; Amazon extended the experience of shopping for books; Google made navigation through the web less hit-and-miss.

It has often been observed that the Semantic Web (SW) needs to gain similar momentum to the Web; Metcalfe's Law [1], which states that the utility of a community is proportional to the square of the number of members, implies that a large network needs to be in place before the utility of the SW will become visible. In that case, SW research may well be aided by informed guesses about what a SW killer app would look like. Research in this area is not just an example of academics trying misguidedly to think like entrepreneurs; rather, the success of the SW as a technology *depends* to some extent on where the killer apps lie.

However, mere innovation is not enough. A killer app must meet a need, or open up some opportunity, for enough people to create a critical mass of users; hence its use must integrate into people's work or other behaviour patterns. And, as well as the social aspects of the user base, there must be sufficient economic advantage for some businesses to take the risk of development and marketing.

In this paper, we discuss the notion of killer apps in the SW, in the context of Protégé. The idea of a killer app sounds very broad, but there is quite a literature on the ways that killer apps transform business contexts. In the next section we will discuss potential features that killer apps are thought to require, and features thought to be desirable, and link these into the current state, and potential, of Protégé, in relation to the SW, and its inevitable needs for ontology management.

2 Killer Apps on the Semantic Web

Spotting a killer app is in reality as much art as science. However, there have been a number of academic and business studies of the concept, e.g. [2][3], which tend to discuss and categorise killer apps by some of the

features they exhibit in relation to their environment; of course, context is very important to judging killer apps, an application that appears at the wrong time or in the wrong context is as good as useless.

In the remainder of this section, we will think about killer apps in the context of both the SW as an environment, and Protégé as a technology applied in that environment. We will discuss some of the features that would be useful for the SW, and map these features onto general features that we might expect to see in a killer app, and onto the current structure of, or potential for development for, Protégé.

Protégé has evolved over many years to become one of the most popular tools in ontology editing and management, with more than 26K registered users at the time of writing. Protégé has certainly become a killer app in its niche domain. For Protégé to reach this level of popularity it must have shown some superiority over other similar tools, such as OntoEdit, Ontolingua, WebOnto, OilEd, and KAON, which all co-existed with Protégé at some point in time. A comparison of earlier versions of these tools is detailed in [4]. Results showed that Protégé has many superior features to most of its competitors at the time, such as open source, freely available, extendible design, friendly user interface, range of plugins offering additional features, excellent technical support, etc. A quick look at the qualities that boosted Protege's popularity may teach us some lessons about developing killer apps in general, and SW killer apps in particular.

2.1 Indispensability

There is nothing better than being indispensable to make one wanted. This is as true of killer apps as it is in any other area of life. Such an app must grow into an essential part of an existing market (e.g. eBay, Amazon) or even create a new market of its own (e.g. Napster). Being first in a new market is a distinct advantage [2][3], but late entrants can also succeed if they outperform existing services [5].

Support for Ontology Development: There is no doubt that ontologies will be one of the SW supporting pillars. The simplicity of Protégé's user interface made ontology development and management a much less costly and challenging task. Manually writing ontologies directly in RDF or OWL is out of the question, and hence a graphical tool to shield the complexity of SW languages is certainly very beneficial.

In addition to editing an ontology schema, ontology developers often need to query it, compare it with other ontologies, extend it with further schemas, populate it with instances, etc. Protégé addressed such needs and expanded its use to not only the edition of schemas, but also of building and managing entire Knowledge Bases (KB).

2.2 Hitting the mark

One important feature that a killer app must have is the ability to meet a range of requirements for a particular task. The technology must also outstrip its opponents, usually in terms of costs, but also by doing the job better, or by providing extra services interleaved with the fundamental task

Cost: Killer apps are often cheaper than similar, or alternative, market products [3]. The more affordable an application is to obtain and operate, the more users will be able to embrace it. Giving an application or a system for free is probably the best cost-effective approach to make them ubiquitous [6].

Superiority: A killer app has to provide higher service quality than other existing technologies to justify its use and attract the masses (e.g. email vs post mail, broadband vs dial-up connection).

Ease of Use: Easy to use, non-complex applications certainly increase the likelihood of their adoption and usage. Imagine where the Web would be now if everyone had to learn HTML before they could even browse it!

Personalisation: Customers tend to become more loyal to services they can customise to their liking [2]. Many of today's killer apps have some level of personalisation. Amazon for example makes recommendations based on what the customer buys or looks at, Auto Trader [1] and Rightmove [2] save customers' searches and notify them via emails when a new result to their query is available. Personalised web services attract more customers (if done properly!) and provide better tailored services [5].

[1] http://www.autotrader.co.uk
[2] http://www.rightmove.co.uk

2.3 Leveraging Metcalfe's Law

Finally, an important task for any killer app is to develop a user community, a large number of users each of whom adds value to the network.

Communities of Practice: A killer app should have the potential to create a community of users and aid this community to grow and develop. Metcalfe's Law states that the utility of a network is proportional to the square of its user base [1], and so a tenfold increase in the number of users, for instance, makes the network one hundred times more valuable. This law is often used to explain the success of many of today's killer apps [2][6][5], and the value of networked applications such as the Internet, email, and instant messaging software. One email account is useless as there will be no one to email! But as the number of email accounts increases, so will the value of this application.

Take Yahoo messenger for example. They made it very easy for existing members to invite others to download and use the tool. They also provide a number of games and chat rooms where members can meet to encourage the creation of more interconnections between their members, thus boosting their network value.

Open Systems: When a system opens to direct interactions with other existing systems, its value increases by acquiring additional value from those systems [6]. By being open source and extendable, Protégé allowed many existing systems and tools to be linked or integrated with it, thus increasing Protégé's use and value. For this reason, and for being available for free, Protégé is quickly becoming the default ontology editing tool.

Active User Community: User community of Protege increased dramatically once it became open source and freely available [7]. This has benefited the tool greatly by widening its user base, thus increasing its network value by many folds.

Protégé supported its large community through emailing lists, short courses, and conferences, providing it with an excellent technical support which has so far been up to scale, another quality of Protégé, missing from most of its competitors.

By going open source, and adopting an extendable architecture, Protégé managed to move part of the development effort to third parties. User were able to build their own personal extensions to Protégé, and many third party plugins have become available to the community. The continuous emergence of plugins is one of the most attractive features of Protégé, which made it stand out amongst other similar tools.

2.4 A range of requirements for the Semantic Web

All of these considerations leads us to think of how the SW in general, and the tasks associated with ontology management in particular, will ideally drive development of Protégé. Essential aspects of the new SW environment with which Protégé may have to grapple include the range of SW languages that have been appearing and will continue to appear, the potential size of the SW, and issues related to the publishing of and access to online knowledge services.

Language Support: Protégé has been quick to endorse new SW schema representation languages, such as RDF and OWL, thus staying up to date with W3C's latest standardisations. With respect to RDFS, Protégé extended the language to enable it to express certain things, such as cardinality, multiple slot range, etc. [8]. Even though those extra representations did not conform to the RDFS standard, but they were necessary for more detailed domain representations. Those extensions are specific to Protégé, and hence can not be read by other RDFS tools.

Scalability: Not a long time ago, much effort was given for building very large ontologies (e.g. CyC [3], Gene Ontology[4], SUMO[5]). As the SW is developing, particularly under the stimulus of the annual Semantic Web Challenge, we are beginning to see knowledge bases containing small to medium size ontologies, but with very large number of instantiations [9]. The winner of the 2003 challenge, CS AKTive Space[10], contained 25m RDF triples stored in 3Store[6], while the winner in 2004, Flink[7], had 35m stored in a Sesame[8] KB. KBs of this order require a great deal of management.

[3] http://www.cyc.com/
[4] http://www.geneontology.org/GO.biblio.shtml
[5] http://ontology.teknowledge.com/
[6] http://www.aktors.org/technologies/3store/
[7] http://prauw.cs.vu.nl:8080/flink/
[8] http://www.openrdf.org/

Protégé's main design goals where interoperability, and ease of use [11]. Scalability was not at the heart of Protégé's design, as is the case with some triple stores, such as 3Store[9], Sesame[10], and Kowari[11]. This needs to be addressed if Protégé is to be used as a platform for launching SW applications. Protégé certainly offers much more than the triple stores mentioned above, in terms of editing capabilities, data visualisation, ontology merging tools, versioning, etc. If Protégé is to deploy services over large KBs, then it has to scale accordingly. RDF and OWL database backends for Protégé are available for dealing with large KBs. The RDF(s)-DB backend plugin that connects Protégé to a Sesame repository is a good effort [12], assuming that Protégé's user interface can also scale to high limits.

Publishing and Access: Providing online knowledge services is a crucial part of launching the SW. The triple stores mentioned earlier are designed for this purpose, with online querying facilities using querying languages such as RDQL and SPARQL. Both of these languages are W3C standards for querying RDF stores. Protégé currently does not have a strong and current support for such languages.

3 Conclusions

We have looked at some important factors in the development of the Semantic Web. Ontology development and management will be an essential task, and tools will be required for this. But for an application to become a killer app for the SW it has to meet certain requirements and possess certain qualities.

We discussed those features and qualities in the scope of Protégé, which managed to become a killer app for ontology editing and management for a large number of SW enthusiasts. By looking at analyses of the characteristics of killer apps in general, and the requirements of the SW in particular (even requirements that strictly go beyond what Protégé is capable of providing), we can get some sense of how Protégé should develop further in the SW context.

Acknowledgments

This work is supported under the Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration (IRC), which is sponsored by the UK Engineering and Physical Sciences Research Council under grant number GR/N15764/01. The AKT IRC comprises the Universities of Aberdeen, Edinburgh, Sheffield, Southampton and the Open University. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing official policies or endorsements, either express or implied, of the EPSRC or any other member of the AKT IRC.

References

1. Gilder, G.: Metcalfe's law and legacy. Forbes ASAP, 13 September (1993)
2. Downes, L., Mui, C.: Unleashing the Killer App. MIT Press, Harvard Business School Press (2000)
3. Christensen, C.M.: The Innovator's Dilemma. Harvard Business School Press (1997)
4. Gómez-Pérez, A., Fernández-López, M., Corcho, O.: Ontological Engineering. Springer-Verlag (2004)
5. Evans, P., Wurster, T.: Blown to Bits. Harvard Business School Press (2000)
6. Kelly, K.: New Rules for the New Economy: 10 Radical Strategies for a Connected World. Penguin Books (1998)
7. Gennari, J.H., Musen, M.A., W.Fergerson, R., Grosso, W.E., Crubézy, M., Eriksson, H., Noy, N.F., Tu, S.W.: The evolution of protege: An environment for knowledge-based systems development. International Journal of Human-Computer Studies **58** (2003) 90–123
8. Noy, N.F., Sintek, M., Decker, S., Crubezy, M., Fergerson, R.W., Musen, M.A.: Creating semantic web contents with protege-2000. IEEE Intelligent Systems (2001) 60–71
9. Klein, M., Visse, U.: Semantic web clahhenge 2003. IEEE Intelligent Systems **19** (2004) 31–33
10. Shadbolt, N., monica schraefel, Gibbins, N., Harris, S.: Cs aktive space: or how we stopped worrying and learned to love the semantic web. In: Proc. 2nd Int. Semantic Web Conf, Florida (2003)
11. Noy, N.F., Fergerson, R.W., Musen, M.A.: The knowledge model of protege-2000: Combining interoperability and flexibility. In: Proc. 2nd Int. Conf. Knowledge Engineering and Knowledge Management (EKAW'2000), Juan-les-Pins, France (2000)
12. Askar, K., Siril, Y., Dougherty, M.: Passerelle - a plug-in that connects protégé to sesame. In: Proc. 8th Int. Protégé Conference, Madrid (2005)

[9] http://www.aktors.org/technologies/3store/
[10] http://www.openrdf.org/
[11] http://www.kowari.org/