Towards a Knowledge-Level Software Platform

A position paper for the International Semantic Web Workshop
Stanford, July, 2001

Avron Barr, Shirley Tessler and Buddy Kresge*

A software platform is, to students of the software industry, what volcanoes are to geologists – dramatic manifestations of powerful subterranean forces, often colossal, sometimes threatening. There have been many software platforms over the years – questions as to their relative importance or goodness must be left to a longer discussion. As the Semantic Web community works out a way to author, encode, interpret and manage machine-readable knowledge that is applicable across a range of systems and subject domains, we may be witnessing the emergence of a major new software platform.

Historically, software platforms have varied widely in terms of their impact on the evolution of computing, measured, for example, by the range of applications that run on them, and thus the number of lives they touch. To become an OS360 or Windows behemoth, a platform must not only embody the right mix of technologies at the right time, but must additionally serve the needs of three disparate constituencies:

- **Solve the real needs of customers.** For example, Windows established itself among PC manufacturers by allowing them great flexibility in components and bus designs, while largely insulating software publishers and corporate application developers from the complexity that resulted. Market share on new and installed PCs in turn attracted hundreds of independent software publishers. Customers care about well-supported software that is compatible with other systems already in place. Products should also reduce customers' operating costs, comply with standards, and have lots of utility (applications), largely supplied by the independent publishers and corporate applications developers.
- **Create opportunities for software publishing companies** and, to a lesser extent, software services firms. Independent software publishers decide whether to publish to the platform – without them you don't actually have a platform, just a tool. Publishers care about market size, primarily, as well as the talent pool, tools, support/training, long-term platform strategy, and fairness – a level playing field. Of course, the platform also creates opportunities for systems integrators and for software developers.
- **Win the hearts and minds of software development teams**. One dimension of the importance of a software platform is surely its acceptance by a large community of software professionals (not just programmers, but the whole team). Software developers, who mostly go where the money is, will additionally be attracted to platforms with minimal stupidity, powerful tools, language-independence, and a rational, if not elegant, design. They also like openness, so that they can build their own tools.

Whatever technology is involved, a software platform results from a complex social arrangement among these stakeholders.

---

* Barr and Tessler are strategy consultants for software companies, software startups, investors, and government planners through their firm, Aldo Ventures. Kresge is a knowledge engineer at Essential Logic and a specialist in the healthcare benefits knowledge domain.

While it may seem premature to inquire about the impact of the Semantic Web on the structure of the software industry, history has shown that the eventual outcome of the platform-formation process is often shaped early in the evolution of the platform. Breadth of vision, choice of partners, openness, and dumb luck are all major success factors.

Who Touches The Knowledge?

The Semantic Web involves a nexus of technologies: distributed computing, HTTP, XML, RDF, frame-like KR, agents, inference engines, and more. To see the direction matters must take to produce a new software platform, consider the following application space in U.S. corporate healthcare benefits (healthcare providers, insurers, corporate HR, and employees):

- Insured employees can inquire about their benefits and options via their company's portal, at work or at home.
- Corporate HR departments can do benefits planning, get bids from insurance carriers, and easily exchange real-time information with business partners, like enrollment and eligibility outsourcers, based on their unique, complex employee group structure.
- Insurers, service providers and insured patients can communicate about claims and bills.
- Healthcare providers can get approvals in real time. Physicians can know instantly which alternative medications are covered by a particular patient's insurance plan.

Until all the players' websites are semantically enhanced, the full value of the shared knowledge cannot be realized. And there are thousands of domains like healthcare benefits, where knowledge-enabled applications make sense. The huge number of different applications and knowledgebases that must be built and maintained by a wide variety of people, make this look to us like the makings of a major software platform: these software developers will want to share tools and infrastructure to "raise the level" of their knowledge-level programming.

In addition to tools for programmers, the effective use of machine-readable knowledge in an industry requires that a large number of knowledge editing tools be created, so that different people in different roles can author, test, share, audit, and maintain the knowledgebases in a natural way. (Out-of-date knowledge is worse than useless.) The platform and infrastructure must support the construction of these tools. It must also support the evolution of abstractions that tie the whole industry together. In healthcare, for example, the concept of "benefit" is associated with a set of services, a cost formula, a payment mechanism, a liability, and so on, depending on one's perspective.

Trilogy Corp., an early success story in the use of explicit knowledge to support e-commerce, uses proprietary knowledge modeling tools, and scores of Stanford graduates, to maintain a single, proprietary knowledgebase of electronic components. For the Semantic Web, thousands of companies will be maintaining similarly complex knowledgebases. The tools must support sharing of industry abstractions and knowledge maintenance by non-PhD's! Eventually, there will likely be several layers of tools – another indication that a software platform might emerge.

Of course, there is a lot of money to be made by companies who enable this next level of e-commerce, where applications can offer increased functionality by accessing companies' published knowledgebases. But the infrastructure that must be built to enable this kind of knowledge sharing may have implications beyond commerce: explicating what we think we all know is a first step toward coming to a common understanding.