Open-Source and the SDL/Idiom Deal

By Frank Bergmann

The SDL/Idiom deal has sparked concerns among industry players who fear data spying and vendor lock-in, among other things. Open-source has frequently been mentioned as a possible alternative, but for many, a question persists: Is open-source ready for prime time?

Market consolidation is a normal process that occurs in most markets: Once the initial growth phase levels off, there's room for the standardization of technologies and business practices. The resulting consolidation means that a larger share of the market is concentrated with fewer companies.

In the localization sector, consolidation seems to favor companies with strong combinations of technology and service offerings. Most of the top 20 LSPs (Localization Service Providers) provide customers with such an in integrated solution. We can assume that only this combination of technology and services creates enough lock-in to allow LSPs to earn the margins they need to grow beyond a certain level. The SDL/Idiom deal seems to be in line with this tendency, allowing SDL's service division in the future to lock out competing LSPs.

This development causes a couple of notable problems: For one, it threatens many second- and third-tier LSPs that don't have unique technologies to offer. Additionally, ITDs (Internal Translation Departments, the buyers of localization services) might see prices of tools and localization services rise as the number of suitable service providers shrinks — along with the pricing variations that a competitive market spurs.

This is where the open-source development model might kick in. Only the open-source model has the power to break up these emerging oligopolies, as we can witness in a number of other software markets.

However, there is no open-source product available today that might compete with the closed-source standard applications. This article will attempt to determine if and how it is feasible to create a full-fledged open-source translation workflow solution. In particular, we will address the question of collaboration in a highly competitive industry and the success factors necessary for such a product to succeed in the marketplace.

Open-Source = Collaboration Rules

It is a common myth that open-source software (OSS) is only about pasty software geeks using their free time to write complex software that no normal user might ever understand or use. While it's true that a large number of projects in open-source communities such as SourceForge.net are of that type, the most successful OSS projects today are structured in a very different way.

Today, the real power of open-source is the creation of a level playing field that allows fiercely competing companies to collaborate in a specific, limited area. The best example of this type of collaboration is the famous Linux operating system. Today more then 50 percent of core Linux developers are directly or indirectly paid by large corporations such as IBM, Red Hat, Novell, Sun Microsystems and others. Still, each of these companies provides their customers with a very different value proposition by combining the Linux kernel with other systems and services.

One important institution in this field is the Open-Source Development Labs (OSDL). The OSDL have been founded by a group of Linux-using companies to coordinate development activities. A steering committee consisting of the paying members determines the overall direction of development. Could this be a collaboration model for the localization industry?

Presenting FOLT

So who or what is FOLT? We can read on www.folt.org: “Forum Open Language Tools (FOLT) is a forum and working group of service providers in the field of translation and documentation.” FOLT was founded in 2005 and currently consists of ten LSPs, including Euroscript, one of the world’s top 10 LSPs. It receives support from several government organizations and universities.

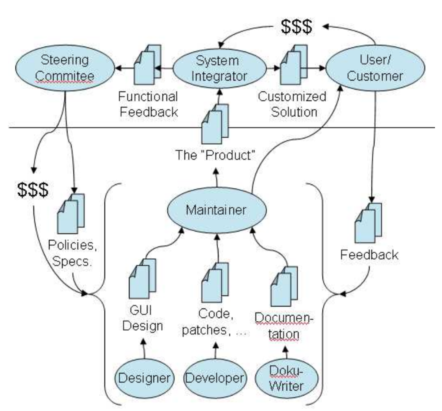
FOLT’s main activity so far has been to start the TMOSS (Translation Memory Open Source System, www.tmoss.org) project in 2007 with the aim to “create a TM (Translation Memory) system as Open Source Software.” Currently, TMOSS consists of some functional specifications for a TM system to be built. FOLT will celebrate an “open meeting” in Germany on the 20th of February; the outcome of this meeting will be worth keeping track of.

FOLT may be a suitable candidate to catalyze wider industry collaboration, though it may not remain the only candidate; other competing initiatives could be founded in the future. Additionally, it is not completely clear whether or not FOLT has sufficient experience with open-source software development to successfully lead such a large-scale project. Even if this does prove to be the case, we can expect a natural selection process to take place, and modified reincarnations of FOLT to continue the work.

What steps should FOLT or another organization take in order to successfully develop commercial-grade software? Several open-source TM systems already exist: OmegaT, Sun Translation Tools, Foreign Desk, and a plethora of tools used for the translation of open-source projects themselves (KBabel, for example). However, none of them has found any major traction in the marketplace so far. Why?

Ecosystem

One important aspect of successful open-source projects is an “ecosystem” that exists to deliver the open-source code to the customer. The main roles of such an ecosystem are:



A schematic open-source ecosystem

The System Integrator: The System Integrator is capable of customizing a product to the particular needs of the customer, and delivers other important services such as training and bug-fixing. The System Integrator role may be filled either by an external IT consulting company or by an in-house IT department.

The Product Manager: The Product Manager makes day-to-day decisions about which features should be included in the software and when it’s ready for release to the public. The Product Manager needs to work closely with customers and the steering committee in order to satisfy often contradictory goals — producing stable software while rapidly incorporating new features, for example. The prototypical Product Manager is Linus Torvalds, the inventor of Linux. Torvalds is accepted as the ultimate authority in decisions related to the Linux kernel, and is the root for a hierarchical network of trust spanning a community of more than 2,000 developers.

The Development Team: The Development Team may consist of various types of contributors, such as paid software developers and members of System Integrators. However, all successful open-source projects exhibit a “meritocracy” (see below).

Steering Committee: [fixme]

Users/Customers: [fixme]

None of the existing open-source TM systems have so far managed to create an ecosystem around their systems. It will be crucial that FOLT and other organizations learn how to involve capable System Integrators and allow them to create value for their final users.

Meritocracy

Another important part of any OSS community is the drive of their members to be recognized as experts. This pattern is frequently called meritocracy, or the distribution of power according to merits.

The benefits of being recognized as an expert are clear: The leading software experts will receive high-paying consulting jobs, and leading experts or companies will be the first to be contacted by potential customers. For this reason, the members of an open-source community will compete to be recognized as experts by answering difficult forum questions, implementing new features, serving voluntarily in the steering committee, and so forth. In short, the meritocracy is what makes an open-source community work. FOLT and other organizations will need to learn to create a meritocratic system in order to be successful.

Technology

Translation memory technology itself apparently isn’t very complex in technical terms. For example, the excellent MetaTexis Web-TM system has been developed by a single part-time software developer. As such, there is little doubt that FOLT will be able to produce a more or less suitable system.

A Web-based workflow system *á là* Idiom is more complicated because of the underlying Web technology. However, the availability of efficient Web development toolkits has greatly reduced the effort needed to build such an application. Additionally, developers could base their work on existing applications in order to reuse their infrastructure. FOLT apparently looks for Eclipse as a platform, so there is a lot of sample code available.

The main challenge may therefore come from the user's side. Most TM users are linguists with relatively low technology affinity. Freelance translators in particular may see a new TM as a waste of time — and money, since their learning time is usually not paid for. Any newcomer needs to be aware of this, and provide a flawlessly working system. This includes support for other parts of the translation environment such as the operating system, Java Virtual Machine, Internet connectivity, and so forth.

With some good luck, FOLT may be able to convince on the of the small TM vendors to open-source their code and to rise with FOLT to industry leadership. Such a deal would speed up the technology development for FOLT, and the small TM vendor may earn much more through consulting then by trying to earn license money with small LSPs.

Integration

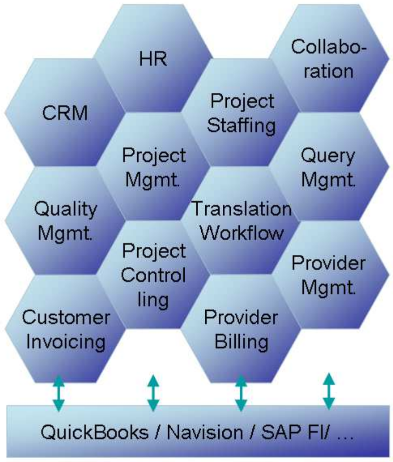
An interesting option to increase the size of the community is to integrate the TM server backend with other applications, such as the Alfresco or Typo3 content management systems (CMS), support and helpdesk applications. It might even be worthwhile for FOLT to allow integration with closed-source applications as Documentum. In this case, FOLT would need to choose a permissive open-source license or provide technical means for the server side to allow closed-source vendors to separate open-source from their systems via interfaces.

On the other client side, it could be interesting to integrate existing translation environments such as the Sun XLIFF editor or the KBabel translation client in order to add a considerable base of existing users to the project. In order to do so, FOLT would need to focus on specifying a protocol between client translation environments and the TM-server application. Such a standard protocol might prove more powerful than any specific reference application.

FOLT and]project-open[

]project-open[is an open-source based Translation Project Management tool. The author of this article is the founder and managing director of]project-open[.

Many people see]po[as a competitor to across and Idiom, and as a possible candidate to fill the "Idiom gap." However, we see little overlap between these workflow systems and]po[. Workflow systems manage the linguistic side of translation, while]po[concentrates on the financial and business side. The currently existing]po[workflow is based on the upload/download of files, and is not integrated with any TM.



]project-open[software modules. Translation Workflow represents only a small part of the system.

For this reason we will be happy to integrate any open-source TM or workflow system with]po[as soon as the first prototypes exist. We ultimately believe in the vision of an integrated OSS translation system covering all important components of the l10n value chain, but we also see that customers may want to use closed-source systems for their own reasons.

Conclusion

FOLT appears to be a suitable candidate for taking the leadership in open-source TM management. However, other competing initiatives could be founded in the future. Natural selection will determine how these initiatives play out. We expect that many ITDs and LSPs will join FOLT in order to counter the concentration in the L10n tool space.