A Profound Experiment in Openness

David Wood

Symbian Foundation
1 Boundary Row, Southwark, London SE1 8HP, United Kingdom
davidw@symbian.org

Abstract

This paper provides an inside view on the experiences and interactions of the open-source Symbian Foundation with the wider mobile community in the 16 months since the announcement of the formation of the foundation. The paper reflects on learnings regarding some profound challenges and difficulties in pursuing a policy of thoroughgoing openness – especially in the context of the transition of a large existing codebase from closed source to open source.

Index Terms: Openness, Open source, Open management, Symbian Foundation.

I. Introduction

Whilst "many hands make light work", it's also said that "too many cooks spoil the broth". This homely wisdom reflects the sharp difficulties facing any collaborative endeavor. When there are conflicting visions about the direction of the roadmap for new mobile products, who decides? How does clarity of purpose and direction emerge from a cacophony of suggestion, debate, and demand?

The Symbian Foundation was born from the conviction that, done right, the open community approach will (over time) generate better solutions than any system of tight control. Taking the spirit of openness to heart, this paper reflect candidly on the progress made so far by the foundation. Openness has brought its own difficulties, as well as its own triumphs. There are many learnings for anyone interested in the best way forwards for the mobile industry.

A. The benefits of openness

For a software system, openness is expected to bring many benefits.

The core idea of openness is to make it easier for more people to participate in building and using that software system. That's opposed to a closed system which is (at one level or another) fixed, unchangeable, or difficult to extend or modify.

More people means "more eyeballs" – more people looking at the problems and opportunities of the software system, and devising new solutions. To quote Eric Raymond, from the Cathedral and the Bazaar [1]:

- "Given enough eyeballs, all bugs are shallow"
- "Given a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the fix will be obvious to someone."

More people also means "more IQ" – a wider range of thoughts, suggestions, and analyses.

Finally, more people also means "more innovation" – a greater likelihood of out-of-the-box thinking, unconstrained by group mind-think.

So there are in principle many benefits to openness, *if done right*.

B. Three levels of openness

During the process of the formation of the Symbian Foundation, benefits were foreseen to exist for each of three levels of openness. [2]

The first level is **open SDKs** (Software Development Kits) – which list and describe APIs (Application Programming Interfaces) that are supported by the software system. The SDK allows the creation of add-on and plug-in software which takes advantage of the software system. For example, applications can be written, taking indirect advantage of the underlying power of the hardware and the networks of a device, without needing to understand the variations of that hardware and these networks across different devices. The software system hides these differences, and allows multiple add-on applications to co-exist.

The second level is **open source** – in which people get to see, not only the result of the software (namely, its APIs) but also *the recipe for creating that software*. Open source brings many additional benefits, beyond those provided by open SDKs. By allowing people to study source code (especially inside a run-time debugger), it can throw light on why particular APIs behave the way they do - sometimes with unexpected quirks, as compared to what would be anticipated from reading the formal or informal description provided in SDKs. It also allows people to copy aspects the software – often resulting in better starting points for their own software. And, critically, open source allows people to alter the way the whole software system works. So they can participate more fully in the grand project, not (only) of *extending or supplementing* the original software, but of *improving* that software. This is innovation at the heart, rather than (just) innovation at the edges.

The third level is **open management**. (See [3] and [4] for related concepts.) Just as open source reveals the recipe to APIs, open management takes the principle one step further, and (in effect) reveals *the recipe to the planned evolution of the source code* (and APIs). The owners of the system publish their intentions in advance, describing the problems they are facing, and the ideas they are considering for addressing these problems. They share their research data and their current hypotheses. They welcome feedback on these ideas, giving plenty time for the community to influence and alter their thinking. Community members are able to monitor and influence the decisions as they're being made. And just as open source (for greatest value) needs it to be clear how to build that source code, so also open management (for greatest value) needs it to be clear what are the governance principles for making decisions on the ideas submitted. Open management, done right, attracts the deepest trust and the best consideration of ideas for improving a software system.

That's the general theory. How does this apply to the Symbian Foundation?

II. Expectations in Creating the Symbian Foundation

First, some history.

A. The need for openness in the mobile industry

Plans for the Symbian Foundation were drawn up at a time (2008-9) of growing challenges facing advanced mobile devices and applications. The challenges include:

- **Usability** – users had problems discovering functionality, and learning how to use it; their experience using mobile devices often contained as much frustration as enjoyment
- **Complexity** – impacting usability, but also impacting both manufacturability and programmability

- **Performance** – issues with battery life, speed of responsiveness, and cost of required hardware
- **Quality and robustness** – software being released before being adequately debugged or optimized
- **Privacy and security** – potential risks with malware and with data loss
- **Business model uncertainties**.

The Symbian Foundation consciously embraced openness at all three levels, believing this to be the best route to involve a sufficiently vast quantity and diversity of people and organizations, who could then address both the issues and the rich opportunities of in the mobile space. [5]

Openness was seen as key for one additional reason – to avoid any one company from attaining a position of control or dominance over the mobile industry. Instead, the leading mobile platform should be owned and managed on behalf of the entire industry.

B. Drawbacks to openness

Of course, there are drawbacks to openness. Many successful companies in the mobile and/or software industries deliberately choose working methods that stop short of the full range of openness described above. For example, both Apple and RIM prefer to operate with much tighter systems of control than the one envisaged by the Symbian Foundation.

The predecessor company of the Symbian Foundation, namely Symbian Ltd, was another company that stopped short of this vision. Symbian Ltd provided an open SDK, but did not adopt open source. The reasons for this decision, at the time, were:

- The business model of Symbian Ltd depended on royalty payments – device manufacturers paid a fee of around $5 to Symbian Ltd for each device shipped based on the software
- The customers of Symbian Ltd were initially apprehensive about widespread use of open source
- Other attempts to use open source in mobile seemed to fall foul of fragmentation pressures, with multiple incompatible branches of software being created – to the detriment of any potential ecosystem that would otherwise provide solutions on top of this software system.

On the other hand, Symbian Ltd practiced something like a half-way step towards open source, sometimes called “gated source”. The source code for Symbian OS was available to a group of companies who were prepared to pay a significant price for this privilege (around $30k per annum) and to sign a substantive legal agreement. The community realized that there were considerable benefits even from this restricted sharing of code – benefits akin to those applicable to open source. However, it remained a fundamental rule that (apart from pieces of example code) source code should not be made public. The business model of both Symbian Ltd and its partners depended on that rule holding.

As regards open management, a similar “half-way house” situation applied (the same as for open source): Symbian Ltd practiced a kind of intermediate step towards open management. Various consultation bodies existed, such as the Technology Committee (usually known as “TechCom”), the Operator Review Board (“ORB”), and the Japan Review Board (“JRB”). Typically, these bodies would meet around two or three times a year. In each of these bodies, Symbian Ltd personnel would share aspects of their forward planning, and receive feedback from experts in the community. However, this dialog was restricted to a small number of participants, and was conducted far away from the public eye. Any such public discussion was shunned:

- It was believed to be likely to be counter-productive
- It would have involved revealing confidential aspects of future roadmaps.

C. Changing attitudes toward openness

During the lifetime of Symbian Ltd, general attitudes towards open source changed.

The shareholders and customers of Symbian Ltd were impressed by the successes of, for example, the Mozilla Foundation (which oversees the user-friendly high-performance Firefox web browser) and the Eclipse Foundation (which oversees the extensive family of Eclipse developer tools). It was seen that open source was moving into the mainstream of computing – especially via the use of "weak copyleft" licenses such as the Mozilla Public License and the Eclipse Public License.

Second, the shareholders and customers of Symbian Ltd reached the view that the revenues that could be earned from licensing the Symbian platform software would be dwarfed by the revenues from selling hardware and services that use this software. In other words, the best financial outcome for them would be to enable the fastest and deepest growth of the capability of the Symbian platform, so that better and more valuable hardware and services could be sold based on that software – rather than seeking to maximize their own share of the software licensing revenues. Making the software free would result in better financial payback:

- Not (primarily) because of less cost being required to pay for license fees
- But because of the additional innovation and development that would be encouraged as a result.

In summary, the potential large upside of allowing and enabling faster and deeper innovation came to be a more significant consideration, in the minds of the shareholders and customers of Symbian Ltd, than the potential downsides of working with open source – especially in the intensifying competitive marketplace for advanced mobile devices and services.

Therefore, widespread planning took place, to devise processes that would maximize these upsides and minimize these downsides. This planning resulted in what can be termed the Symbian Foundation open management process.

D. Intended open management process

Here's a brief introduction to the intended open management process of the Symbian Foundation.

First, as much management discussion as possible should take place in the open. As far as possible, there should be no restriction on who can see the discussion or participate in it. Private mailing lists should be avoided. For example, discussions on roadmaps, strategy, architecture, usability, and so forth, should take place in public mailing lists and/or web discussion forums.

For an example of open discussion, take a look at the Symbian Ideas Exchange Marketplace site at ideas.symbian.org. This site is described [6] as "being developed as a place where people can:

- "Propose and assess ideas about Symbian (technology, services, strategy, initiatives…)
- "Weigh up the community's best thinking about Symbian
- "Guide Symbian management in decision making
- "Establish reputations as leading thinkers about Symbian
- "Participate in Symbian's philosophy of Open Management."

Second, people who demonstrate particular expertise should be elevated into roles in which their opinions have greater authority. For example, on the Ideas site, people can be designated as "experts" for a specified category of idea, with greater weight being given to their votes in that category.

The same principle underlies the operation of the Symbian Councils. The Councils vote on any matters of potential large change in the Symbian platform software. To quote from [7]:

- The **Feature and Roadmap Council** invites proposals for contributions from the community and seeks to coordinate new contributions into a unified platform (or tools) roadmap
- The **Architecture Council** invites and reviews technical solutions for new contributions in order to ensure the architectural integrity, backward compatibility and fitness-for-purpose of enhancements to the platform
- The **User Interface Council** invites and reviews descriptions of new user interface elements and develops guidelines to help ensure high quality device user experiences
- The **Release Council** coordinates the integration of contributions into stable and timely platform and tools releases.

The stated hope [8] is that, in most cases, a council will be able to reach consensus decisions. On occasion, votes will be necessary. This raises the question: who gets to vote?

Although everyone is able to track the discussions and proposals for council meetings – and is welcome to provide opinions – in the end, it is a smaller number of member companies who have a voting responsibility. These companies are selected onto Council seats by the Symbian Foundation Board of Directors, out of the set of companies that indicate their interest.

Council participants must be signed members of the Symbian Foundation. The criteria to be considered by the Board to determine membership include the following:

- Which council(s) the member in question is being proposed for, and the rationale for the choice of council(s)
- Details of any contributions made or intended to be made to the community by the member (these may include code, resources, participation in working groups, co-marketing activities, and so on)
- Details of the application, service or device shipments represented by the member's offerings, including the portion of those currently and potentially based on the Symbian Platform
- The names and biographical details of the proposed primary and deputy representatives
- Details of any particular expertise held by the member relevant to work undertaken by the council.

Smaller scale changes in the Symbian platform are decided by individuals known as "package owners". The entire software system is divided into around 130 distinct packages – each typically having at least 100k lines of code. The small-scale evolution of these packages is in the hands of the nominated package owner, who reviews which contributions from the community to accept, and which to reject. Individuals can become package owners following a period of time as a contributor to that package. Changes of package owner (when moving from one company to another) are reviewed by the Architecture Council.

III. Practical experiences in creating the Symbian Foundation

That's the theory: how are things working out in practice? In practice, openness brings challenges, as well as successes.

A. Problems experienced: open SDKs

Even the discipline of open SDKs (the first level of openness) brings challenges:

- An API that exists for one version of the software system ought, in general, to also exist for a later version of the software – and to behave in the same way. This is known as the principle of compatibility. Without a compatibility guarantee, developers have to exert much greater effort in making their add-on or plug-in software work across multiple devices
- The requirement in a new version to preserve compatibility in a previous version imposes significant constraints on the development of the new version. A design that turns out to be suboptimal often cannot simply be discarded, but must be worked around
- In turn, this raises the requirement for greater skill in designing APIs, so that these APIs can be "future proof" against unexpected new requirements – but without introducing too many extra layers of software redirection and abstraction, which can significantly slow down run-time execution
- There's also a requirement for classification of APIs. For example, "public" APIs will in general be preserved at all costs, and "platform" APIs are more liable to change between versions (and application developers, therefore, are discouraged from using them).

The good news about this particular set of challenges is that they are already reasonably well understood. Symbian Ltd built up considerable expertise in API management and compatibility control over many years, before the formation of the Symbian Foundation.

B. Problems experienced: open source

Open source brings its own challenges too. If community members are able to produce their own versions of the platform software, the community quickly ends up with multiple different branches (also known as "forks") of the system. The license may stipulate that all these different changes are made available (as open source) for the owners of the platform to re-integrate. However, that doesn't mean the different changes will be easy to integrate. Instead, the different branches will often each embody different assumptions and simplifications, which are not appropriate to be accepted (without further change) into the mainline. This is summarized by the saying that "fragmentation is easy, integration is hard". [9]

Again, Symbian has a past track record in dealing with this issue – from the days in which Symbian platform software was spread through a (relatively small) grouping of customers and partners. In these days, customers and partners frequently produced local variants of the Symbian platform. However, over time, the Symbian community developed expertise in:

- Preparing customer-specific changes in ways that made them easier for the mainline system integrators to evaluate and review
- Pushing out improved solutions in subsequent releases of the platform, which removed the need for selected customer-specific changes that had previously been necessary.

C. Problems experienced: open management

The area where perhaps the most novel challenges have arisen is that of open management.

An open discussion about the future roadmap of a technology area depends on the package owner sharing his or her own ideas about the future of that area. However, there are drawbacks to this kind of sharing:

- Confidential details about as-yet unannounced products may be inferred from these proposals. ("Ah, the package owner is going to be introducing protocol XYZ in the release in June next year, that means that a phone from that company coming out shortly later will probably have feature WVU…")
- The package owner company may fear that they will lose their competitive advantage if news of their future plans becomes externally known
- The package owner company may be unwilling to enter into further discussion about the merits and demerits of a particular proposal, taking the view that there is no value to such discussion and no leeway in their own plans to entertain further changes.

Even though senior management of a company may express support for open innovation, individual middle managers of that company may all reach the view that they want to retain an element of secrecy and competitive advantage in their own local domain. Unfortunately, whilst these individual decisions each make sense in their own right, they add up, overall, to opposition to the principle of open innovation.

Discussion of future roadmap items can become particularly difficult in times when a company is under "closed reporting" restrictions (for stock market purposes). Financial or legal managers in the company may object to continuing dialog about technology features, when that dialog could be interpreted as assuming various future market scenarios.

In principle, the solution is for companies to separate any intended "secret" new contributions in two:

- They include some enabling API in an existing platform module
- They write the actual new functionality in a separate module, whose details are not described until later.

A more general reflection is that asking for feedback can easily be misconstrued. See [10] for a discussion of this point.

Ridicule is just one of the hazards of an open conversation. Another risk is that you frustrate people, who think that a matter should already be decided, and there's no need to keep the conversation going. You can also hurt people by failing to censor comment responses (in a blog) that have all the appearance of being destructive, spiteful, trolls. Yet another risk is that, in adopting a brainstorming approach, you will say something tentatively and/or provocatively that gets (wrongly) interpreted by readers as a new statement of corporate policy. For example, a statement such as “if xx and yy don’t happen, then it might be best that zz happens” can be taken out of context and misinterpreted as “Symbian official spokesperson wants zz to happen”. Woops.

D. Problems experienced: cultural collisions

Openness also faces problems from cultural clashes with elements of the mobile value chain where tight control is perceived to be necessary.

For example, network operators (with some justification) are distrustful of allowing users to download and install any application they choose, onto their mobile devices. The risk is that some of these applications will interfere with the performance of the phone:

- Flattening the battery too quickly
- Preventing the user from making phone calls in some circumstances
- Running up higher than expected phone bills
- Copying information from the user’s contact list.

The result of open access to applications could, therefore, be expensive service support calls, and an unfair decline in the user’s assessment of the network operator (“Why did they sell me this phone which doesn’t work?”). It is to avoid this problem that the Symbian Signed program was introduced. Symbian Signed is designed to balance the needs of, on the one hand, ease of writing and deploying applications, and on the other hand, preventing malware. This is a difficult balance to get right. The tension underlying the balance is heightened by an overt promotion of open source and open management.

This is not the only example of a cultural clash which still needs to be worked through. Other clashes include the remnants of the “free software” versus “open source” dialectic, as represented by the “strong copyleft” versus “weak copyleft” licenses. It remains awkward and legally controversial to combine software written under a strong copyleft license with software written under a weak copyleft license.

E. Problems experienced: transitional effects

Finally, the transition from closed source to open source has introduced challenges of its own:

- The previous Symbian OS offering contained components which were licensed (for a fee) from third parties. Since these components were not owned by Symbian, they had to be left out of the open source Symbian platform offering, resulting in so-called “Third Party IP holes”
- The IP owners of other parts of the software did not wish the software all to become open source immediately, seeking more time to complete security audits and other checks of the source code. This resulted in code being, for a while, released under a new license, the Symbian Foundation License, ahead of its eventual release under an open source license
- Lack of low-cost reprogrammable reference hardware further reduced the initial attractiveness of the software system to would-be hackers or researchers.

As more of the components are released under the Eclipse Public License [11], and as low-cost reference hardware becomes available [12], these challenges will, presumably, become less significant.

IV. Concluding Reflections

It's still very early days for the Symbian Foundation. No doubt as more time passes, additional examples will emerge of challenges and opportunities for the Symbian Foundation philosophy of openness.

The Symbian Foundation can already be seen as fertile ground for researchers and software practitioners alike, to weigh up the potential effectiveness of various implementations of openness.

Some questions whose answers will presumably become clearer in the months and years ahead include:

- What is the best way to determine whether a given decision can be taken locally (for example, by a package owner) or if it should be reviewed by a council?
- What is the best way to ensure that councils act as agents of positive contribution rather than (just) as agents of stating requirements?
- In what circumstances is it appropriate to develop secret new functionality on top of an open source platform, and in these cases, what is the best way to achieve this result?
- What is the role for application approval in a world of mobile open source and open management?
- Would it be advantageous for any one open source license to come into a position of greater dominance for mobile software? Are the existing open source licenses the best that we can expect the industry to adopt?

Acknowledgment

The author would like to thank Professor Joel West [13] of San Jose State University, and Lawrence Simpson [14] of the Symbian Foundation, for very helpful feedback on earlier drafts of this paper.

REFERENCES

[1] http://catb.org/esr/writings/cathedral-bazaar/cathedral-bazaar/ar01s04.html
[2] http://telephonyonline.com/wireless/news/symbian-foundation-non-profit-0408/index1.html
[3] Joel West and Siobhán O'Mahony, "The Role of Participation Architecture in Growing Sponsored Open Source Communities", Industry & Innovation, April 2008, http://www.joelwest.org/Papers/WestOMahony2008-WP.pdf
[4] Siobhán O'Mahony, "The governance of open source initiatives: what does it mean to be community managed?", Journal of Management and Governance , May 2007, http://www.springerlink.com/content/c229131m5v22271g/
[5] http://developer.symbian.org/main/about/going_open_source/index.php
[6] http://symbianpathways.com/2009/08/23/looking-for-producers/
[7] http://developer.symbian.org/main/about/councils/index.php
[8] http://blog.symbian.org/2009/03/14/incremental-platform-development/
[9] http://dw2blog.com/2008/06/18/the-dangers-of-fragmentation/
[10] http://symbianpathways.com/2009/08/18/the-challenges-of-open-discussion/
[11] http://blog.symbian.org/2009/07/08/security-package-in-epl-tick-whats-next/
[12] http://developer.symbian.org/forum/showthread.php?t=2816
[13] http://www.linkedin.com/in/joelwest
[14] http://www.linkedin.com/pub/lawrence-simpson/0/100/1b5