When Innovation is Hard

Does the Open Source Development Model Work in Hardware?

ilkka.tuomi
meaningprocessing.com
Theses

1. Software and hardware are epistemologically different artifacts
   - This leads to different innovation and conflict models in software and hardware projects
   - The success of the open distributed approach in "close-to-hardware" software development (e.g. Linux kernel) is partly explained by this epistemological difference

2. Close-to-hardware software, social software, and hardware are different
   - Close-to-hardware software development is easy to distribute because the developer community consists of a homogenous network of people and tools
   - There exists a "phenomenological bottleneck:" Real hardware can only be approximated using digital representations; distributed hardware projects can therefore be really hard
   - There also exists a "political bottleneck:" Social software can be hard as there are multiple interpretations and systems of meaning
Different Types of Animals Require Different Styles of Breeding

- Close-to-hardware SW
  - In a given HW environment:
    SW is both the description and the implementation
    Mapping between software and the functionality of the system is one-to-one
    Technical functionality can be empirically tested

- HW
  - The context is the open world:
    Unmodeled features matter
    Systems wear, tear and break down
    Although dominant voices are loudest, evaluation criteria have to be negotiated from multiple perspectives

- A homogenous developer community, rooted in its own "objective reality."
- A homogenous developer community, rooted in its own "professional reality."
  or:
- A network of interacting communities, each with their own stocks of knowledge.
The Epistemic Difference

- Close-to-hardware software
  - Approximates well the positivist and empirist assumptions

In the domain of SW, only those things exist that are explicitly described

Meta-level questions (e.g. progress) can be answered by empirical tests

- Hardware
  - Operates under the constraints of phenomenological ontology and epistemology
  - Approximates well the social constructivist assumptions
The Study Method

• Study open hardware development projects that are very similar to open source software development
  • but where the object of development is a physical product
  • Baseline: Linux kernel

Software that is closely integrated with the underlying processor hardware
  • Reference pools:

SourceForge.net software projects in the ”Linux,” ”Hardware,” and ”Finance” categories
  • Physical products: semiconductor cores at OpenCores.org

*Semiconductor cores*: pre-designed building blocks (processors, DSPs, bus-controllers, ...) that can be combined and re-used to build integrated circuits, including systems-on-chip.
The Case: OpenCores.org

- "SourceForge" style portal that focuses on sharing logic designs (semiconductor "intellectual property" cores / virtual components)
  - Developers can use their preferred license but are encouraged to use GNU or BSD-style licensing
  - OpenCores.org provides a CVS version control system that enables developer collaboration
- Founded in 1999 by Damjan Lampret, Slovenia
- 308 projects in CVS (Sept. 2008)

```vhdl
entity OR_ent is
  port( x: in std_logic;
        y: in std_logic;
        F: out std_logic);
end OR_ent;

architecture OR_arch of OR_ent is
begin
  process(x, y)
  begin
    if ((x='0')
```
The Study Domain: "Chipless" Semiconductor Producers
## Top IP Core Vendors

<table>
<thead>
<tr>
<th>Rank</th>
<th>Company</th>
<th>Employees 2007</th>
<th>IP Revenue ($M) 2007</th>
<th>Market Share 2007</th>
<th>Cumulative</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ARM</td>
<td>1728</td>
<td>516</td>
<td>29.6%</td>
<td>29.6%</td>
</tr>
<tr>
<td>2</td>
<td>Rambus</td>
<td>430</td>
<td>180</td>
<td>10.3%</td>
<td>39.9%</td>
</tr>
<tr>
<td>3</td>
<td>Synopsys</td>
<td>5196</td>
<td>97</td>
<td>5.6%</td>
<td>45.4%</td>
</tr>
<tr>
<td>4</td>
<td>Motorola-TTPcom</td>
<td>286</td>
<td>87</td>
<td>5.0%</td>
<td>50.4%</td>
</tr>
<tr>
<td>5</td>
<td>MIPS</td>
<td>196</td>
<td>83</td>
<td>4.8%</td>
<td>55.2%</td>
</tr>
<tr>
<td>6</td>
<td>Mosaid</td>
<td>112</td>
<td>57</td>
<td>3.3%</td>
<td>58.5%</td>
</tr>
<tr>
<td>7</td>
<td>Silicon Image</td>
<td>635</td>
<td>51</td>
<td>2.9%</td>
<td>61.4%</td>
</tr>
<tr>
<td>8</td>
<td>Virage Logic</td>
<td>417</td>
<td>47</td>
<td>2.7%</td>
<td>64.1%</td>
</tr>
<tr>
<td>9</td>
<td>Imagination Technologies</td>
<td>366</td>
<td>43</td>
<td>2.5%</td>
<td>66.5%</td>
</tr>
<tr>
<td>10</td>
<td>SST</td>
<td>715</td>
<td>40</td>
<td>2.3%</td>
<td>68.8%</td>
</tr>
<tr>
<td>11</td>
<td>CEVA</td>
<td>192</td>
<td>33</td>
<td>1.9%</td>
<td>70.7%</td>
</tr>
<tr>
<td>12</td>
<td>Chipidea</td>
<td>310</td>
<td>33</td>
<td>1.9%</td>
<td>72.6%</td>
</tr>
<tr>
<td>13</td>
<td>ARC</td>
<td>196</td>
<td>29</td>
<td>1.7%</td>
<td>74.3%</td>
</tr>
<tr>
<td>14</td>
<td>Mentor Graphics</td>
<td>4358</td>
<td>25</td>
<td>1.4%</td>
<td>75.7%</td>
</tr>
<tr>
<td>15</td>
<td>Wipro-Newlogic</td>
<td>350</td>
<td>21</td>
<td>1.2%</td>
<td>76.9%</td>
</tr>
<tr>
<td>16</td>
<td>Dolphin Integration</td>
<td>164</td>
<td>17</td>
<td>1.0%</td>
<td>77.9%</td>
</tr>
</tbody>
</table>
Many Small Firms
OpenCores.org
According to Alexa

Daily Reach (percent)
opencores.org  kernel.org

Visitor Countries
## OpenCores.org CVS

### Table:

<table>
<thead>
<tr>
<th></th>
<th># developers</th>
<th>lines of code</th>
<th>lifetime (months)</th>
<th># commits</th>
<th># revisions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>mean</strong></td>
<td>1.4</td>
<td>43,211</td>
<td>9.6</td>
<td>28.4</td>
<td>182.5</td>
</tr>
<tr>
<td><strong>max</strong></td>
<td>37</td>
<td>1,501,619</td>
<td>160</td>
<td>1712</td>
<td>12363</td>
</tr>
<tr>
<td><strong>min</strong></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>median</strong></td>
<td>1</td>
<td>4,273</td>
<td>1</td>
<td>4</td>
<td>29</td>
</tr>
</tbody>
</table>

N= 308
Projects by Code Size

![Projects by Code Size Graph](image-url)
Number of Developers

More than 5 developers: 6 projects (1.9 %)
More than 1 developer: 39 projects (12.7 %)
SourceForge vs OpenCores

Wilcoxon-Mann-Whitney, alternative: true location shift is not equal to 0
data: OpenCores and SF – hardware developers
W = 128623, p-value < 2.2e-16
Project Developers
Close-to-Hardware, "Social SW", and Hardware

SF – financial excludes an outlier with 393 developers
Developer Distributions

![Graph showing cumulative distributions for different sources.

### Table: Cumulative Distributions

<table>
<thead>
<tr>
<th>Source</th>
<th>Min.</th>
<th>1st Qu.</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Qu.</th>
<th>Max.</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenCores</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1.41</td>
<td>1</td>
<td>37</td>
<td>308</td>
</tr>
<tr>
<td>SourceForge - financial</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2.19</td>
<td>2</td>
<td>393</td>
<td>2435</td>
</tr>
<tr>
<td>SourceForge - hardware</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3.09</td>
<td>3</td>
<td>65</td>
<td>1263</td>
</tr>
<tr>
<td>SourceForge - Linux</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3.26</td>
<td>3</td>
<td>84</td>
<td>827</td>
</tr>
<tr>
<td>SourceForge - financial (ex. 393)</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2.02</td>
<td>2</td>
<td>79</td>
<td>2434</td>
</tr>
</tbody>
</table>

© I. Tuomi 2 Oct. 2008 page: 17
OpenCores are Different

Wilcoxon-Mann-Whitney, alternative: true location shift is not equal to 0

data: OpenCores and SF – hardware developers
\( W = 128623, \text{ p-value } < 2.2e^{-16} \)

data: OpenCores and SF – Linux developers
\( W = 87228.5, \text{ p-value } < 2.2e^{-16} \)

data: OpenCores and SF – Finance developers
\( W = 311118.5, \text{ p-value } = 9.675e^{-10} \)

data: SF - Linux and SF – Finance developers
\( W = 1164896, \text{ p-value } = 1.110e^{-15} \)

data: SF - Linux and SF – Finance (ex. 393) developers
\( W = 1164896, \text{ p-value } = 8.882e^{-16} \)

data: SF - Linux and SF – Hardware developers
\( W = 515923, \text{ p-value } = 0.606 \)
Conclusions

• Most open source projects have a small number of developers
• SourceForge projects are different from OpenCores projects
  OpenCores projects have significantly fewer developers
  • The results are compatible with the thesis:
    Close-to-hardware software is easier to develop in an open distributed development model than hardware
    • Hardware-related software also seems to be easier to develop than "social SW" (e.g. accounting, ERP, ...)
    This is compatible with the thesis:
      – When there is no shared ontological basis, development becomes difficult
    • When project outputs are not "tightly-coupled functional systems," different dynamics are possible
    e.g., WikiMedia, representative democracy, ...
Discussion

• The size of developer communities (number of people committing code) is different for close-to-hardware, social software and physical hardware projects.
  
  • Does this reflect difficulties in coordination and collaboration? Or something else? (e.g., the number of potential competent developers)

• Is open source development innovation or problem solving?
  
  • Is this ”open innovation,” or ”distributed collaboration”?

• Assume the downstream multi-focal innovation model (Tuomi, 2002):
  
  • ”Innovation happens when social practise changes.”
  
  • ”There are always several stakeholder communities, each with their own systems of meaning and measures of progress.”
  
  • ”An innovation artifact is often a 'carrier' of multiple innovations that have only loosely coupled evolutionary paths.”
  
  • ”Innovations are always retrospectively described, usually from the dominant point of view.”
  
  • ”Mainstream innovation histories fill the gaps and invent facts to make the story convincing.”

• Is the translation from abstract functional specification to a physical device a process that can destroy interpretative flexibility? (e.g., by sedimenting the innovation in a way that makes it difficult to reinvent and misuse it?)