

# gitFAB : A web service for sharing repositories of open hardware for duplicating and forking

Daisuke Akatsuka<sup>†1</sup>, Hiroya Tanaka<sup>†2</sup>

## 1. Background

Digital fabrication tools such as 3D printers and laser cutters, and electronic prototyping platforms such as the Arduino have recently emerged and numerous physical works such as chairs and desks or electronic devices have been proposed. For many such items, the source code, parts, tools, blueprints, circuit diagrams and construction logs are published openly in a manner that resembles the culture of open source software on the Web.

One reason that open source software has proved popular is that the source code and the know-how it embodies can be easily accessed, modified, duplicated and forked allowing it to be refined, re-used and re-purposed. In open source hardware, sharing of blueprints and construction logs is common. However, duplication and forking have not received much attention. Unlike software, hardware is composed of various physical components. It requires assembling and fabricating to duplicate. As a result, construction notes are of great importance and, in order to facilitate duplication and forking, such notes must not only be readable but also encourage reusability.

Blog services such as Instructables and Thingiverse can be used to share blueprints, source code and construction documents. However, using such a service, the writing style is not constrained and the writer may use any style and structure they wish to express the flow and steps of construction. Since there is no standard means for representing these materials they are difficult to duplicate and fork. On the other hand, by using a consistent approach to representing this material, not only is the readability for humans increased but so is the potential for reusability by machines. For example, a machine may be able to output the STL data to a 3D printer automatically. In the future, collaboration of humans and machines may be seen in the DIY scene as well.

Formalizing the format of construction data is one answer to managing the incompatibility of project materials. However, it is difficult to find a single format that accommodates all applications. Also, by limiting expression to a single format, the diversity of projects may suffer.

## 2. Case study

FabSource<sup>i</sup> of FabLab Japan is a sharing service primarily aimed at craft projects made by 3D printers or laser cutters. FabSource can share resources freely, such as blueprints, project descriptions and construction logs. Common information such as the project category and the tools and materials used, are managed as a database. However, the writing style is still dependent on the writer; there is no system to ensure consistency.

Instructables<sup>ii</sup> is web site to share DIY projects including not only craft projects but also electronic devices. The writing style takes the format of a step-by-step log. In addition, the service can also handle resource files.

Thingiverse<sup>iii</sup> is a service to share 3D data produced by MakerBot inc. Since this data is intended for 3D printers, a construction log is not required.

Hometalk<sup>iv</sup> is service for DIY projects that do not use digital tools. It is primarily made up of photos of projects, occasionally augmented with blueprints and construction logs.

In common to all of the above services is the aim of sharing resources and construction logs. Unlike these services, our project, gitFAB, is particularly focused on duplicating and forking.

## 3. gitFAB

gitFAB is a web service that aims to facilitate duplicating and forking for hardware. To this end, the service aims to improve the readability and reusability of project materials whilst also accommodating free expression. As such it features a free writing style with a consistent format.

gitFAB has a function to upload resources such as blueprints, images, and source code. In addition, document editing is performed on a paragraph-by-paragraph basis. Each paragraph can be labeled with a heading which marks the different items of the project. For example, suppose there are several projects that describe a clock. The common headings from the different project descriptions could be inferred as the minimum items required for making a clock.

With regards to forking, a user can duplicate the project and change only the blueprint, source code, or one of the paragraphs. As a result, the project documentation takes a bottom-up approach with gitFAB providing a common format. We also note that not all makers of DIY are skilled document writers. However, since gitFAB allows users to modify just a single paragraph, the skill required is much lower and the burden of documentation much lighter thus encouraging more sharing and more making.

Finally, logs are written using markdown syntax which allows a large degree of freedom in expression.

---

<sup>†1</sup> Mozilla Japan

<sup>†2</sup> Faculty of Environment and Information Studies, Keio University

#### 4. system

gitFAB includes functions to manage projects such as adding, updating and deleting content as well as functions for duplicating, forking, and uploading resources. These functions are similar to the git<sup>v</sup> version management system which is popular for open source software development. gitFAB is built using the github<sup>vi</sup> repository service as a backend. gitFAB manages a project as a git repository. The repository is managed under the user's github account by using github's OAuth support and resource files are stored in the git repository. The project document is stored as the README.md file in the git repository so a user browsing the github repository can easily read the project documentation.

It should be possible to search gitFAB projects by tag however github does not provide this function. In order to address this limitation, meta-information such as tag information is stored in the README.md file. In addition the gitFAB server tracks this metadata in a database for searching. In a sense, github is the hard disk and gitFAB is the memory. If the gitFAB server crashes or is unavailable, it is possible to recover the data from github.

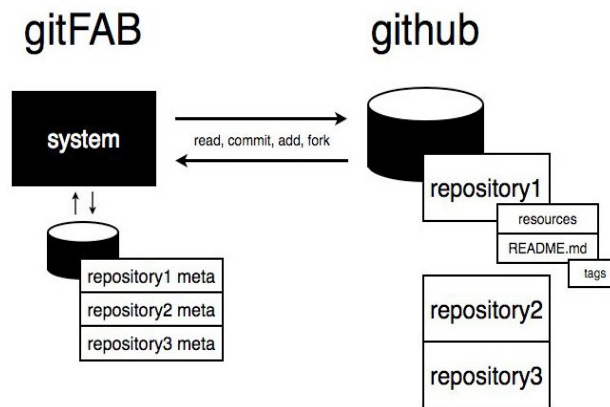


Figure 1. System architecture

Another feature of gitFAB is the facility for to select between two different browsing modes: document mode and presentation mode. In gitFAB making documents for Web sharing or for presenting are one and the same. The user can modify the layout and colors of the document by editing a CSS stylesheet.

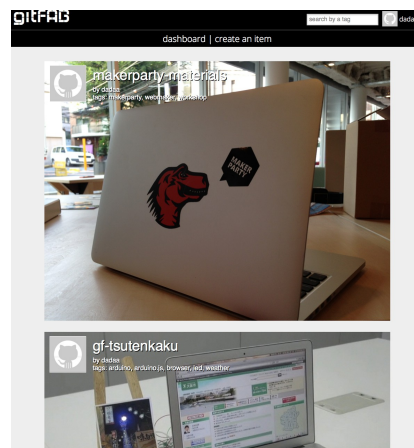


Figure 2. Screen shot of list of projects



Figure 3. Screenshot of the document

## 5. Future work

Recently, although many workshops have been held to encourage building open hardware, the contents of such workshops are rarely shared. Our initial target will be to facilitate sharing the contents of such workshops. We hope this will make it easier to produce similar workshops and therefore increase opportunities to share the know-how and techniques of digital fabrication and building electrical devices thus resulting in even more interesting and sophisticated open hardware.

Furthermore, if logistical information about the workshop such as the number of participants and date is stored in the document, it may be possible to duplicate and fork the workshop. This same approach may be suitable for school homework or a software workshop. There may be many other activities that are suited to gitFAB which we have yet to imagine.

Finally, we want to use the data in gitFAB to determine what are the most suitable formats, rules and approaches for open hardware development.

- i FabSource <http://fablabjapan.org/recipe/>
- ii Instructables <http://www.instructables.com/>
- iii Thingiverse <http://www.thingiverse.com/>
- iv hometalk <http://www.hometalk.com/>
- v Git [http://en.wikipedia.org/wiki/Git\\_%28software%29](http://en.wikipedia.org/wiki/Git_%28software%29)
- vi GitHub <https://github.com/>