Decentralized Version Control and Mass Collective Collaboration in Design

A Case Study of a Web Application Utilizing the Diff Algorithm and Automated Design Generation

Yasushi Sakai¹, Daisuke Tsunoda²
¹,²Nikken Sekkei ltd.
¹,²{sakai.yasushi|tsunodad}@nikken.jp

Especially in early design phases, there is still potential to achieve collective design in architecture. To investigate the possibilities of mass collaboration, this study is based on a web application (http://lmnarchitecture.com) which implements the same technology that present software development stands on. Within those plans, the system calculates the resemblance between the models utilizing an algorithm that computes the difference between texts. The system requires the users to choose one model that is existent, and calculates the resemblance in real-time. As a result, a tree diagram is collectively achieved each having the link of inheritance. Two types of "Bots" (automated plan generation) was implemented to examine the effect of human-machine collaboration. As a result, there were 1750 models submitted in three months. Throughout the models created by humans, in this system, 49% of the models were inherited by the same user, and 23.04% of the models inherited bots.

Keywords: Democratization, Collective modeling, Design automation, GIT

INTRODUCTION
It is obvious that we humans attained knowledge by learning, which is essentially the stacking of experiences of the past. Additionally, ever since science has achieved its method, the accumulation of knowledge has bolstered us to discover and innovate new ideas. One technique was to illustrate the chain of inheritance in knowledge is to use a tree diagram. Although the work of Architecture is fully collaboration involving different kinds of specialties with a wide range of consults, the field still has room for investigating potential of systematical inheriting. Architectural tradition shows that designers still work independently, without a clear intention to exchange ideas in design. This view can be derived from what architects uses as tools for design. Attempt to collaborative spatial design has followed since the advent of the internet, yet modern CAD systems still lack in decentralized version control which is crucial to multi user participation.

On the other hand, software development is rapidly changing its process influenced by the movement of social networking. Today numerous open source programs are developed by the collaboration
between vast varieties of people who never meet physically. As a result, the evolution of software resembles the form of nature having different versions and functionality. Moreover, software development can be interpreted as a scientific method where each study increments its knowledge balancing citation and new ideas. Studies achieve reputation from the number of citations as much as the innovativeness. To integrate this accumulation of knowledge in the field of spatial design, it is necessary to measure the difference between alternatives, and clarify what was added and deleted.

This study tries to unleash the power of collaboration and enabling remix of design by literately showing a tree diagram. Not only enabling a new way of evaluation by providing the diagram itself, the achieved data structure has potential for a new subject - development of automated design intelligence. Supported by today's explosion of data, data mining and machine learning is now the trend of various fields of science.

PREVIOUS STUDIES

Methods of collaborative design
Multi participation in design has long been one goal in architects. Before the advent of CAD systems, architects have relied on physical models to gain feedback from multi individuals. However, the process of how the model was generated is hard to reproduce and have decreased its number by cost. Historically, the relations and influence among different architects and their work has long been tacit. The "Open Building" idea will lead in theory, were "users/inhabitants may make design decisions as well" introduced from Habraken. Other methods such as classification of design methods into patterns Christopher Alexander (1977) is another approach for inclusive design.

Internet and methods borrowed from software development
The defiance can be observed in the early days of intercommunication progress, starting from the collaboration via internet and video chat. Attempts followed using a virtual reality atmosphere, such as online conversation using Second Life. As Rosenman (2007) points out, there are still "issues of multi representations of objects, versioning, ownership and relationships between objects from different disciplines."

One study from Burry (2005) using SVN in the process of design, another version control system widely used in the context of software development combined with Gehery Technologies' Digital Project, share the same intent. The study argues that direct utilization of SVN was not possible within binary data, and took an approach of file locking method which simulates the process. To circumvent the binary issue, for this study, the data structure was included in the development of the web application. The scalability to consumer software remains, yet gains control of the version controlling process. File locking procedure can be seen in modern BIM software giving functionality of multi-user participation. However, the file locking procedure is limited to centralized version control. Only one model can be edited in one time and restricts users to work concurrently in one design. In other words centralized version control creates a hierarchy throughout the versions, this aspect is suitable to later phases of architectural design, where effective accumulation is what we needed to crystalize the proposal. Opposed to SVN, this study’s version controlling mechanism is "decentralized". Since the proposals have no hierarchy, it is easier to have gain interactions between versions.

Functions of modern CAD and BIM software
In the world of software for designing, Autodesk's Vault has the capability to track versions in a file level, while the mechanism rely on file locking, essentially a centralized method that hierarchy exists. Moreover, Robert Aish (2000) argues that the concept of "files" affect the project to be fragmented, and negates holistic perspectives to perceive the project. To meet this, modern BIM software has the functionality of multi-user participation within the project, yet Vault and these BIM software share the centralization of de-
sign documents. Again, centralized methods are said to be effective in the later design phases for sophistication of a single design plan that is ready to be developed. However, such hierarchies if they are too rigidly adhered to, may inhibit the effective flow of information in collaborative design. (Cumming, 2005)

Another approach, Autodesk's Fusion has the closest to this study, partial components can be version tracked and exposed to the internet. Fusion presents the possibility of collaboration not only inside projects, but within projects. The distinction with Fusion and this study is integration of the Diff algorithm which handles the similarity within alternatives (versions). While Fusion compares design by recording every user operation, the web application for this study compares models by data. This enables to calculate the distance between given designs, which is crucial to analyse, sort and integrate.

**SYSTEM AND FLOW**  
**Process of User**  
The system holds two aspects of method; the technical software architecture side and the social architecture side that takes account of the incentives of the participants. There are two activities for the users, model creation and evaluation. To be a user, the system requires to register and sign-in to this system. After logging-in, the system gets the last time of each activity from the user database. This time interval constrains excessive operation that will destruct the system. The evaluation needs 10 seconds to recover, while creation requires 3 minutes. There are three major views: tree diagram, model evaluation, and model creation. Figure 1 shows the first view, the tree diagram, that any visitor will see when she accesses lmarchitecture.com.

The tree diagram has a link to every single model, directing to the evaluation view (Figure 2). The evaluation view shows a basic information of the model along with the 3D representation, this view enables users to give point to the model by clicking "add points".
Figure 4
Plan creation view. Color blocks indicate the function of the volume.

Figure 5
Diagram of Diff algorithm.
Each "add points" operation will prepare 10 points to the model and their parent models. The points are affected by the "similarity ratio", which returns a part of the points depending on how similar the model is to its inheritance. This reimbursement of the points is a recursive process until the points are lower than the value of 1. Figure 3 shows that recursive process of model evaluation. No valuation basis is presented by the system, users evaluate by viewing the model.

The "make plan" button directs the user to the interface to create models shown in Figure 4. The system forces the user to select a base model to inherit. The model creation is the place where users are stack different types of blocks of function and create a new model. While modelling, there is one constrain that the total cost of the model should not exceed 30,000,000 yen. Each type of volume has different cost.

Additionally, if the model has two or three stories, the total cost is multiplied by 1.2 and 2.0 respectively. The budget constraints and moderates the models while controlling the solution space to make popular models that is credited by evaluators. The "similarity ratio" is concurrently calculated, using the "Diff" algorithm, introduced in the next section. After the modelling process, the user gives a unique name to the plan. The model attains points from the parent model corresponding to the similarity ratio (=inborn points), and saves the image and adds the data to the database.

**Software process**

As mentioned above, the system is a web application, accessed by WebGL compatible browsers. The system uses "Django", a web-framework based on the programming language python, and database storing the information.

The client side (browser side) is developed in JavaScript, using three major libraries. The first one is "Three.js", which handles the 3D rendering of the models when observing and editing the model. The graphical representation uses WebGL, which is increasingly gaining attention for visualizing 3D elements. 3D modeling has long been a professional skill, from the barrier of the difficulty and the cost of software. Yet for this study, the user will only need a browser to participate to this web application. The user interface was designed to be simple as possible, limiting it to block stocking. The second library is "Data.js" that is used to visualize the tree diagram on the main view. The third library is Google's "Diff-Patch-Match", the Diff algorithm calculates the subtraction within the two models. The algorithm takes two strings and compares it to get what was added and deleted throughout the revision. The system utilizes this algorithm to achieve the distance by getting ratio of the amount of altered data identical data. (Figure 5)

**Automated model generation - "Bots"**

Apart from the human user participation, the system holds two types of method to propagate models automatically using previously submitted models. They can be interpreted as "bots", alike to what we have in the internet to collect data and link information relevance known as search bots.

One type of bot is based upon randomness adding and deleting cluster of function blocks (=rooms). This bot is limited intelligence, since there is no strategy in creating a better model, which is wholly dependent in the whole systems natural selection. (Figure 6 Left)

Another bot uses the technique of "crossing" known in the field of genetic algorithm, sampling different parts from two models and combining them together. (Figure 6 Right) This bot is capable of looking up the top model and the second model to patch work the plan. This way the bot reflects the current
Figure 7
The overall tree diagram that was submitted to the web application [accessed 29 April 2015]
trend and bolster what is popular. With the human participants and bots, the entire system collectively searches the best model.

Figure 8
Magnified view of network of plans.

Figure 9
Top rated plans, 'jaldabaoth' is responsible of auto-generation of plans.

Figure 10
Inheritance tendency of models created by users (human)

RESULT
There were 1750 models submitted to the database within 3 months.

Figure 7 shows the overall tree diagram submitted to the web application. Figure 8 shows a partial image of the tree diagram each connected to their parent (inherited) model. Stroke thickness indicates the similarity ratio: the stronger it resembles the parent.

16 users including one account as BOTs was registered to this system. Figure 9 shows the ranking within the experiment by models.

Another aspect to investigate is the relationship between the parent and child model. The result shows that within the models that was submitted by human, 49.7% of the model inherited the same user. (By themselves)

DISCUSSION
This study, questions the present system of competition in architecture practice, which in this case, the evaluation process and its result is distributed in a sequence of contribution of searching the solution space. This study shows that strategy to do so is necessary to multi-user collaboration in design. Neither of the Bots had a learning routine, meaning lacking strategy across plans. Although, this can be said that bots had no intelligence to learn, the result showed a spontaneous strategy to overcome the human competitors. Unlike human participants, the automated process has no limitation in time, in other words the cost for producing one model is nearly zero for machines. The machine has potential to effectively produce accurate models, once the appropriate leaning mechanism implemented. Figure 10 indicates the chart of inheritance within man made models, which can be interpreted the ratio of collaboration. Throughout this experiment, approximately half of the users used models of their own, in other words, modelers are willing to collaborate with different individuals near 50%. Moreover, the result shows that in a blind situation, humans are open to work together with bots (Table 1). It is clear that collaboration in design still needs further exploration, and the result of this study needs constant iteration to measure the will and potential of true collaboration. However, this study constructed one example of a complete system of collaboration in design.
CONCLUSION

Summary
This study conducted an experiment of collective, collaborative design method using a web application that implements a technology popular in the field of software development. With a renowned algorithm called “Diff” and the simple volume stacking app combined, 16 users signed up for creating 1750 models between three months. An evaluation method was concurrently running, making it possible to add points to the desired model once in an interval. The points are exchanged corresponding to the similarity between the parent and child (inherited) models. Within the submitted models, two types of bots (automated design generation) was introduced to collaborate with human participants. Motivation for the (human) participants was controlled using a nested competition system, where users are given the chance to gain money when the whole system earns some prize. 49.74% models that was modeled by humans inherited their previous models, and 27.23% inherited models from other humans, and 23.04% inherited models that was produced by an automated process.

Further exploration
While not immediate, the method of implementing distributed version control to the process of design has potential to both professional designers and non-proper individuals. The system that was introduced in this study has few elements that is scheduled to progress. From the aspect of human participation, crowd sourcing the generation will be more efficient to collect design. Services such as Amazon’s "Mechanical Turk" will be suitable for distributing work to the mass. As the data produced by human aggregates, there will be more opportunity for machine to learn. The possibility of implementing machine learning will be the second field for further investigation. Other than including non-proper individuals, application to consumer software for professionals is one path of evolution. Although the data complexity is much higher and dense then the system we have developed, investigation is necessary to implement to tools that is utilized in practice. A relatively straight forward step is to integrate this idea to a visual programming environment, such as the Grasshopper plugin of a 3D modeller called McNeel’s Rhinoceros. Specifically, Grasshopper files has an xml-like data structure, which is easy to apply Diff and other operations to analyse the data. BIM software have the same potential, since the data structure is easy to be interpreted as database structure. Alternatively, the input data can be a physical (clay) model for to blend into the flow of early stages of architectural design, yet it will limit only to geometry data.

REFERENCES
Aish, R, 2000 'Collaborative Design using Long Transactions and “Change Merge”, eCAADe 18, pp. 107-111
Cumming, M. 2005 'Distributed and Constructed Knowledge in Design Education', eCAADe 21, pp. 501-504