Predicting the Future: Open Source CAAD?

Kostas Terzidis[1], Jan Jungclaus[2]
[1]Harvard University Graduate School of Design,
[2]Harvard University Graduate School of Design
http://www.gsd.harvard.edu/faculty
[1]kostas@gsd.harvard.edu

This paper will present a prototype open source CAD system developed recently by a join effort among Harvard, CMU, and MIT. The system is composed of an expandable user interface, a data structure that supports 2D and 3D objects, image processing capabilities, animation, network communication (TCP/IP), serial interface, and file processing modules that can be expanded. The idea was to develop seed modules that can interact with one another in order to be modified, expanded, or new ones added. The language used is Processing and the setup is made to be implemented in an open source format (i.e. GNU and Google Code). The system is an open source universal architectural CAD system that will hopefully serve as the software standard for education and practice.

Keywords: *Open source; digital design.*

1. Introduction

In the last twenty years, computer modeling and animation systems have become an essential part of the architect's education and practice. Recent advancements in building technology have necessitated their use as instrumental in a way that enables architects to address, resolve, and control the increasing complexity involved in the form and function of buildings, structures, and urban developments. As a consequence, a number of software companies have been competing over the last twenty years to establish each one's software as the predominant tool to be used in both education and practice. While the process of design development in architecture has not changed significantly in the last twenty years, the software packages that address that process change constantly. The reason for this discrepancy is that most software systems used in architecture today were not conceived originally for architectural practice but instead were adopted and adapted from other practices such as aeronautical, automotive, naval, or product design. As a consequence, the capabilities, behavior, organization, or price differ significantly amongst software applications creating confusion over which software is to be used where, when, how, and at what price. Further, this problem carries over into the pedagogy of architecture where students are expected to learn as many CAD applications as possible many of which may become obsolete by the time that they graduate.

2. Preamble

A large portion of architectural design is based on geometry as a means of expression for architectural ideas. Most design applications are essentially tools for manipulating, organizing, and implementing

geometrical forms, i.e. lines, surfaces, or solids. While geometry does not change significantly over time, each new version of design software changes constantly. These changes do not reflect necessarily changes in geometry per se but rather they are improvements in the robustness of geometry or the friendliness in interaction of the user with the geometrical space. Nonetheless, today research and development behind CAD systems has reached a point of universal standardization, despite the plethora of software packages: almost all CAD systems today do the same thing in different ways. It is therefore apparent that if there was a common and universal CAD system for architectural design it would eliminate the need to educate, purchase, learn, use, and update software within education and practice.

One of the main reasons behind software updates is the so-called improvements of the source code. As code remains secretive within the software company, users have to wait until the programmers detect, fix, and re-release the next version in order for them to enjoy the changes. While knowledge of programming was originally a privilege of computer scientists, recently an increasing number of architectural students are computer programmers and are in a position to understand, fix, or add new features if they had access to the source code (Seletsky, 2006). Such an emerging trend can be seen in various open source initiatives that allow users to be in control of the software that they use and at the same time to be able to work in a community-like fashion improving not only productivity, efficiency, and communication but also engaging in ideas and theories (O'Reilly, 1999). It is therefore apparent that if there was an open source universal CAD system for architects it would have contributed significantly to the collective spirit of the profession both as a practical and as a theoretical endeavor (Vallance et al.).

3. Development

The project started with an initiative for testing the grounds for the realization of the open source project that included questionnaires, interviews, discussions, and the development of a set of specifications. However, during the discussions conducted with various experts, software developers, faculty, and students an important issue was raised almost unanimously related to the project. While everybody agreed on the usefulness of such a system many had been concerned with the initial phase of the project. How do you start? Is it a collective effort from the beginning or an individual takes leadership initially? Is it based on previous CAD work or is it a brand new project? Everybody agrees that once the initial seeds are laid then people will start contributing to the framework (Koch, 2004). This problem is quite common in open source projects and while one may think that the term "open" implies collective behavior it also presupposes a definition as to what is and how to establish the notion of "openness". Someone needs to define what is open and how it can be implemented. Developers such as Linus Torvalds (Torvalds, L. 1999), Larry Wall (Larry et al), or Ben Frey are such people that set the initial seeds for Linux (http://www.linux.org) Perl (http://www.perl.com) and Processing (http://processing.org). In this case, after conversations with William Mitchell (MIT), Ugo Gagliardi (Harvard), Mark Gross (CMU), David Skok (open source developer), Daniel Greenwood (lawyer, open source), and many more a decision was made to act upon this in the following manner:

1. Use of existing languages and code. Instead of reinventing what is already known, we decided to re-organize, invest, and add on to what is already there. So, after a series of tests, we found Processing to be the best language to be used for the project. In fact, processing is already an open source language, a scripting language based on Java which is also open source. As a graphics-oriented language Processing is not only easy to use but, most importantly, can incorporate easily existing codes and algorithms since it is based on C/C++. Further, Processing has already an active open source developers'

community that may attract developers out of the university circle.

2. Production of an initial CAD open system. We developed a framework for the development of a CAD system. It is composed of an expandable user interface, a data structure that supports 2D and 3D objects, image processing capabilities, animation, network communication (TCP/IP), serial interface, and file processing modules that can be expanded. The idea was to develop seed modules that can interact with one another in order to be modified, expanded, or replaced with new ones (a suggestion by Ugo Gagliardi). The language that was used is Processing and the setup was made to be implemented in open source format (GNU in Google Code) (Stallman, 1999).
3. Documentation of the source code, internal representation, and specifications. This led the first author of this paper to write up a 300-page document that describes the various workings of this CAD system. It covers the data structures used to represent CAD objects, the methods used to process information, as well as the theory upon which the decisions were made. As a side effect, this effort led to a proposal for a book. It is anticipated that the book with its corresponding code will be available to the public by Fall of 2007. The latest code and text can be found at: http://code.google.com/p/oscad/
4. Publicity of the initiative. This has been accomplished with participation in conferences and symposia as well as through invited lectures. In the past year (2006) papers have been presented at eCAADe06, ACADIA06, Sigradi06, ISAIA06, two articles published at formZ newsletter and the journal Design Issues (MIT Press) and lectures were given at Cooper Union, Keio University (Japan), Seoul National university (Korea), Sung Kuyng Kwan (Korea), Inha (Korea), CUHK (Hong Kong), Tonji (Shanghai), Siao Tung (Shanghai) and Tsinghua (Beijing). In all visits the possibility of open source and developed interest, publicity, and links for future development was discussed extensively.

Next Phase

This project is about the development of an open source universal architectural CAD system that will serve as the software standard for education and practice. The system will be developed, maintained, and evolved through collective efforts from within the architectural community under the open source paradigm (http://www.linfo.org/open_source.html). It will be free of charge, open source, and available over the Internet. It will be open to collective technical and theoretical improvement, criticism, and development. It will establish a new standard for education and practice and serve as a prototype for other professions. As an open source system it will allow architects/programmers to experiment further with software and/or scripting.

The role of administration in this project is to establish the theoretical, technical, organizational, and communicational framework for this enterprise. As the founder and principal investigator of the program the first author of this paper is responsible for setting up specifications, making collaborations, and to overview its progress. This phase of the open source CAD system is comprised of the following activities:

- First, the task is to receive feedback on the specifications established so far. While the author's knowledge on the subject is quite extensive and despite the input from other colleagues so far, there is always possible loopholes, mistakes, alternative ways, or missed opportunities. With the development of seed code and a book it is easy now to seek feedback. In this phase, the book with its code will be distributed among a list of experts in order to be criticized, challenged, and modified in an attempt to find a common ground that everyone agrees upon. However, it is important to mention here that this process is significantly faster and more efficient because there is a document and a system already available. The process here is not to create but rather to critique.

- Next, the development phase will start. This will be accomplished initially with the establishment of a GNU/Google code network for mutual code contribution, sharing, and control. Initially, in this phase, the participants will be trusted individuals that will test, debug, and rehearse the communication system. Once the main communication is establish the system will be open to external contributions, i.e. a repository, along the lines of www.sourceforge.net, which will facilitate the initiative by making a forum available for participating parties to post information about new open source code they are contributing to the community and to facilitate other collaboration.
- Next, the tasks will be distributed for development to students, faculty, staff, or private parties as courses, assignments, research assistantships, grants, or contracts. Once a critical mass is developed, the project will self-regulate, evolve, and adapt as an increasing number of architects, programmers, and practitioners will have a vital reason to keep the project going. The basic idea behind this project is relatively simple: when architects/programmers can read, redistribute, and modify the source code, the software evolves: people improve it, people adapt it, people fix bugs. At the same time, this system will function as a universal CAD system, distributed free of charge, open to change, modification, and evolution. The hope is that it will become a standard for the practice, aspire as a living example of a distributed, collaborative, and a universal effort and, as such, may become a paradigm for other practices to follow.
- Once the project is initiated, grants will be sought through national, federal, state, or local foundations. As stated earlier, the aim of such projects is "the good of the community". By supporting such efforts, government agencies will also benefit as there will be less need for standardization of software, redundant training on multiple systems, conflicts of interest between companies and users, and more emphasis will be placed towards collaboration, equal opportunity, and unification.
- Finally, it is anticipated that the mere announcement of such a system will produce a high interest by architects, students, and software developers. The publicity of such an event will be well justified as it will affect significantly the education and practice of architecture. Such publicity will continue throughout the life of the project and further beyond and it will generate interest to the general public as it will function as a political, business, and educational model. In addition, presentations at conferences and publications at journals will engage, inquire, and challenge the pedagogical, philosophical, and commercial foundation of hi-tech architecture.

Benefits

As discussed earlier, the effects of an open source CAD system are expected to be enormous, profound, and far-reaching. If implemented, it will affect significantly the education and practice of architecture as well as the CAD industry. The benefits of such a system can be summarized in the following categories: theoretical, political, commercial, academic, and practical. While certain of the benefits are indeed obvious within the transformation of architectural education and practice, there are certain hidden benefits that need to be pointed out. First, the emergence of a collective system for design invites theoretical discussions on the nature of collaboration, universality, and distribution of knowledge. Second, the development of self-governing and regulating community of software developers is a political antithesis to the current commercial status quo with profound repercussions in the architectural software market. Third, the emerging paradigm shift will gain publicity and set a new direction for other commercially dependent practices (DiBona., Ockman and Stone, 1999). Fourth, academically the communities involved in this project will benefit together with the students,

faculty, and staff as more jobs, assistantships, teaching methods, curricula, and methodologies will be introduced. Finally, the practice of architecture will benefit as a unified free CAD system will replace a fragmentary, costly, multi-content, and foreign to the practice of architecture commercialization.

"Open source CAD software is an idea whose time has finally come. For twenty years it has been building momentum in the architectural culture. Now it finally has the opportunity, means, and support to emerge as a collective effort breaking off from the commercial world, and that will be changing all the rules" (Lakhani and Hippel, 2003).

References

DiBona,C., Ockman, S. and Stone, M., 1999, Open Sources: Voices from the Open Source Revolution, O'Reilly.

Koch, S.: 2004, Free/open source software development, Idea Group Publishing, New York.

Lakhani, K. and Hippel, E., 2003, How open source software works: "free" user-to-user assistance, Research Policy, 32(6), pp. 923-943.

Larry, W., Schwartz, R. L. and Christiansen, T., 1996, Programming Perl, O'Reilly & Associates, Sebastopol, CA.

O'Reilly, T. 1999, Lessons from open-source software development. Commun. ACM 42, 4, p. 34.

Seletsky, P.: Questioning the role of BIM in Architectural Education: A Counter-View Point. [web page] Aug. 2006; http://www.aecbytes.com/viewpoint/2006/issue_27.html. [Accessed 7 June 2007].

Stallman, R. 1999, The GNU Operating System and the Free Software Movement, in: Open Sources: Voices from the Open Source Revolution. O'Reilly & Associates. Also available at URL: http://www.gnu.org/gnu/thegnuproject.html.

Torvalds, L. 1999. The Linux Edge. Commun. ACM 42, 4, 38-39.

URL: http://processing.org.

URL: http://www.linfo.org/open_source.html.

URL: http://www.linux.org/.

URL: http://www.perl.com.

Vallance, R., Kiani, S. and Nayfeh, S., Open Design of Manufacturing Equipment, Paper presented on the Open Design of Manufacturing Equipment at the CHIRP 1st International Conference on Agile, Reconfigurable Manufacturing. p.10. Also available at URL: http://www.opendesign.org/ODF_whitepaper.html.