Web Based Collaborative Architectural Practice Using a Fractal System

Sabu Francis  
Practicing architect, India  
http://www.sabufrancis.com  
sf@sabufrancis.com

I have been working on an architecture representation system in India since 1991; that markedly deviates from the need of traditional drawings as we know. Over three million square feet of work has been done that took advantage of this system as it was being developed. The system has now matured sufficiently to be put into practice as a comprehensive architectural system of practice. It takes advantage of creation of just-in-time dynamic multi-organizations that can get formed (and dismantled) over the Internet on a project to project basis. The raison d’être of the representation system is that it would expose the “source-code” (metaphorically) of any work of architecture to stakeholders, much the same way as an open-source software project exposes the internal representation to fellow developers. I believe the design of architecture must go through an “open source” process in order to produce socially responsible designs. Such a stance is explained in this paper. The paper also explains the system in detail; its mathematical basis and justifies the need for such an approach. It also explores how a collaborative practice can be put into place using the system in the context of Internet technologies.

Keywords: Collaborative practice; fractals; representation system.

Introduction

The central premise of this paper is that architecture can now be practiced in a web based collaborative environment using a fractal system. This is based on a simple discovery in the mathematics of architectural representation that was presented by me to the Journal of Indian Institute of Architects in 1991. Subsequent to that, I wrote a software and tested out the efficacy of the representation system in much of my architectural practice.

The following thoughts are sequentially presented to reach the aforementioned central premise:

a. Architecture can be represented as fractals  
b. Fractals handle complex information, simply  
c. Practice of architecture involves a lot of meta-data defined cooperatively
d. Cooperation for knowledge building, is best done via the web

e. An open-source movement is the need of the day; in architecture

f. In the end, I shall layout the current status of such a web based collaborative practice.

**Fractals in architecture**

Lindenmayer and Prusinkiewicz demonstrated presence of fractals in the representation of plants and other natural forms (http://algorithmicbotany.org/papers/abop/abop.pdf). I observed in India, that even man-made forms could be represented as fractals (Francis, Sabu: 1991). I believe this solves a crucial problem in architectural representation; which is the figure-ground confusion. Resolution of that confusion allows a much richer representation of architecture through the entire process of design.

Architecture contains both solids (built matter) and spaces. They both need to be represented non-redundantly. An architect can represent a space only when it is set off by some built-matter around it. However, the representation of built-matter comes about because there is a need to recognize the space it helps represent. So the representation of one is dependent on the representation of the other. This leads to recursive confusion.

I believe it is critical that in any inquiry, the “object” of focus should be delineated from its background. Unfortunately, architectural design process requires vacillation of attention between solids and spaces.

Conventional modelers and CAD programs often avoid the confusion by only modeling the built-matter. And I think they believe that what is not built need not be represented at all because ... *it is not built*. Some modelers represent both spaces as well as built-matter, but by using redundant representations. Such an approach was not acceptable to me (and other researchers in this area).

There have been successful attempts at modeling both solids and spaces using topological data structures. The “split-edge” topological data structures can handle both built matter as well as spaces (voids). (http://archmedia.yonsei.ac.kr/pdf/collab_with_kalay.pdf)

However it does not respect the growth of a design from hazy initial stages to the final constructed stage; as it demands that both spaces and built matter be simultaneously represented. That would hardly be the case when a design is being fleshed out in a design process. A split-edge topological data structure is valid for the final product, but not for a design as it evolves.

**The SFA (Sabu Francis Associates) fractal system**

In 1991, I proposed that all of architecture be represented as spaces. The assumption being that even solid matter is obtained when one “pours” built matter into a conceptual spatial envelope. An L-System type of grammar can be visualized where the architect “replaces” some of the spaces at a particular scale of examination with finer definitions (sub-structures) of architectural elements in the next scale of examination. This neatly follows the progress of the design process. I believe a recursive-descent grammar can be written using this system, where the terminal tokens are architectural elements that are not further queried for sub-structures (http://en.wikipedia.org/wiki/Recursive_descent_parser).

In this system, spaces are classified as Atoms, Envelopes and Connectors. This classification encompasses all types of spaces. Built matter is clearly representable as the outcome of a boolean operation on these 3 spatial types (envelope minus (atoms+connectors)).

Built matter is also classified into 3 types: Artefacts, nodes and leftovers. Artefacts are solids that have a social semantic (e.g. a table). Nodes are a set of artefacts that collectively form a network (e.g. A plumbing network). Leftovers are the crucial link between spaces and solids – they form space after the aforementioned boolean operation into which
solid matter at that particular scale of observation is “poured”.

When going to the next scale of examination of the architecture, an artefact or a node can again be broken down into a set of atoms, envelopes and connectors. This is similar to the manner in which a camera lens works. The user is free to decide at which scale of the fractal the piece of architecture is to be examined. This also makes it extremely conducive for delegation of work between “generalists” and “specialists” in any design organization.

Fractals & complexity

Any fractal representation covers the holistic at each scale of examination. At the larger scales, a lot of ground is covered; albeit crudely. At smaller scales, a lot of detail is given but for a smaller set of objects. This is similar to the manner in which a camera lens works. The user is free to decide at which scale of the fractal the piece of architecture is to be examined. This also makes it extremely conducive for delegation of work between “generalists” and “specialists” in any design organization.

The SFA system allows the use of simple Euler volumes (volumes that do not have holes in them). There is absolutely no need to represent architecture using Euler-Poincaré volumes. Hence the BREPs (Boundary Representations) used for the geometry in a SFA based software would be a lot simpler than what is found in conventional CAD/BIM software. Neither is the need to tie down the design into strict topological structures such as the split-edge database. In the SFA system, I believe there is smaller chance of combinatorial explosions due to handling of large architectural data.
Using the SFA system, resource intensive boolean operations often need not be carried out. Often, only the intent of the boolean operation need to be stored in the representation. For e.g. in order to calculate the wall area, all one has to do is to subtract the area of the \textit{atoms+connectors} from that of the envelope. The actual boolean subtraction of the geometries need not be done.

The SFA representation allows the most complex of architecture to be represented as a systematized tree of spatial elements and their operations. No detail is ever lost, and there is no preconceived notion of the type of architecture which can be represented.

The SFA system represents both spaces and solids (built matter) non-redundantly. Early stages of design will see a predominance of spatial elements; whereas in the later stages of design more built-matter would be represented. And on construction and post-construction; the representation would be predominantly of built-matter.

The SFA system fits gracefully into the entire time-line of a design: from inception to construction and beyond.

\textbf{Meta-data also required}

Fractals, by themselves, only make the geometrical representation easier and convenient to dig into. Geometry is ONLY one aspect of an architectural element. A lot of meta-data needs to be further associated with each element. Much of the meta-data is of a fluid nature. They cannot be determined beforehand.

Current CAD/BIM software developers put in meta-data fields a-priori. End users don’t get chance to add to those fields; except by intensive customization. The changing nature of architecture design process demands insertion of a lot of just-in-time meta-data into the design model. Users require their own \textit{“name-space”} (to borrow a jargon from computer software \texttt{http://en.wikipedia.org/wiki/Namespace}) which will contain the meta-data tags required to describe categories of issues they face while designing. For e.g. a designer in India may want to have a name-space for specifying issues dealing with Vastu (an ancient system of architectural analysis that some clients in India believe in). That may not be of any use to non-Indian architects who may not have such concerns.

The meta-data can also cover semantic ontologies where the design intent of the project, the activities within etc. are also captured for future use and analysis.

The need today is for a design software system where part of the development process should be in the hands of the user in a friendly manner. Local control should rule over global pre-set fields/rules. I have often used CAD/BIM software only to discard them when I run up against the limits put in the kind of meta-data that was envisaged by the developers of those systems.

As the SFA system puts architectural elements into a systematized tree structure, it is possible to develop software that uses OOPS concepts for handling the representation. OOPS (Object Oriented Programming Systems) allow a very efficient method of applying meta-data into object representations.

I wrote the software in such a manner that the user is not coerced into adopting a predetermined approach to designing. For e.g. It is possible for designers to talk about openings first before talking about the walls into which the openings can go into. The system allows a very abstract and fluid way of designing. The reason for taking such an abstract stance was explained in an earlier paper of mine (Francis, Sabu: 1999, eCAADe Proceedings at Liverpool).
The software “TAD Designer Lite” has been in use, as it evolves, from 1990 onwards. Over 3 million square-feet of work in India have already been done using this representation system.

Static data in any representation contributes only to a limited knowledge. Making the data dynamic and adding meta-data to it makes the knowledge rich. However, creating the meta-data itself using dynamic processes would make the knowledge of the subject even richer. This means that meta-data should ideally arrive at the very last moment so that the decision makers have ample opportunity to consider all sides of an issue.

Applications such as emails and Instant Messengers have demonstrated advantages of transfer of information just-in-time. Web 2.0 technologies such as use of “tags” allow non-linear multi-faceted access to a lot of information.

The web can be used a two-way process: It need not only be used to query for information just-in-time, but meta-data can also be written just-in-time. I believe the correct model for a design practice is to have an open design framework using representation systems such as the SFA system, which is receptive to just-in-time meta-data pulled in from central repositories on the Internet. The information in those repositories can be separately and cooperatively discussed for their quality of content.

Collaborative practice via the web

It is now possible to have collaborative practices which use the SFA system for their design representations. It should be possible for architectural practices to exchange data and meta-data via the web and include them into designs just-in-time.

Figure 4
This shows some abstract calculations required by a municipality. This is done at the initial stage of design.
Thus there would be another (metaphorical) layer of network on top of the Internet; where rich architectural information would flow to and fro.

TAD (The Architect’s Desktop), a software written by me centrally uses the SFA representation system is now being made available free of cost. It can be connected to the Internet for exchanging information with other stakeholders. Using TAD, the entire design process can be examined and enriched. Architects and other professionals can be guided along the process. Currently criticisms are restricted to the product emerging at the end of the design process.

So far, architects have worked in separate metaphorical *islands of knowledge*. The central reason is that the dynamics of the process did not have a representation system to act as alphabets to record the process. Often architects expressed vague fears against any objective representation during the design process.

My experience shows that fears about the design being influenced by the representation system are fallacious: It is like stating that alphabets would limit poetry.

**Open source: Need of the new world**

Today the whole world has become a village and we are facing a lot of pressing concerns. Many of those get even more complicated because the design processes used in the AEC industry are kept away from close scrutiny.

The issues include (not exhaustive): global warming, energy usage, sustainability, toxic waste, sick-buildings and so on and so forth. It is now accepted that one reason for complexity is due to sensitive dependence on initial conditions. For a designer, initial conditions mean the start of the design process itself: Which is where accurate representation is needed.

Figure 5
*This shows the 3D view (including interior) of a small row-house during the final stages of the design.*
The practice of architecture essentially involves the creation of a multi-organization dynamically. The multi-organization is set up between stakeholders to support an architectural project; just-in-time. And it gets dismantled once the project is over. However, the knowledge gained must be archived and retrievable. Accountability must also be possible.

FOSS (Free and Open Source Software) has taught us how to make good products using a cooperative process.

When the source code (aka: the representation of a problem) is revealed to fellow developers during the process of development, there is a marked improvement in the quality of the end product. Important papers such as “The Cathedral and the Bazaar” (http://www.catb.org/~esr/writings/cathedral-bazaar/) have convinced me that development of software in a closeted atmosphere will NOT produce good software.

I believe this to be true even in architecture: It has become critical to expose the process of designing to fellow designers. I believe architecture can be practiced today very analogously to how software is developed by open source developers. Multi-organizations can be set up and dismantled just-in-time; as the need arises. A central system of storage of knowledge \((data+meta-data)\) can be developed using the web in order to archive knowledge properly and promote accountability.

**Current status**

The software required for this project is almost ready and web sites are being put into place. (http://www.teamtad.com) Proof-of-concept such as objective querying of various parameters of a design has been carried out successfully.

This includes an energy assessment application done as a research project with IIT (Indian Institute of Technology; Mumbai). However a fair amount of work still needs to be done; many of them involving the entire fraternity of AEC professionals. For e.g. XML Standards for meta-data transfer need to be cooperatively discussed and set.

**Advantages**
- The fractal nature of architecture offers many advantages in objective analysis. This results in direct improvements.
- Hitherto resource intensive applications such as embodied energy analysis should become much simpler.
- I am confident that the concepts of the SFA system can be extended to give a calculus (akin to those used in L-Systems). This calculus can be used to describe existing architecture mathematically and hence; it should be possible to compare spatial qualities of one piece of architecture with respect to others.
- It may be possible to use the SFA system as a generative grammar; much like the way Lindenmayer used his grammar to explain/explore the growth of many natural forms. Such a generative approach should be explored separately. It has not been my priority so far.

**Disadvantages**
- Any open source system would require a lot of participants. Building up a critical mass of users can be a thankless task. It has many logistic hurdles. Hopefully, the concept of “viral-growth” seen in many Internet projects may help overcome this issue.
- This form of practice is a paradigm shift from the AEC industry’s current practices. It implies re-tooling a lot of software that architects and allied professionals use.
- It has repercussions in the printing industry, use of paper drawings, delegation of work at a construction site etc. This can create serious social issues; especially in labor intensive countries such as India.
- Connections to other open-source representations of architecture have not been considered. A true open source system will never rely on just
one background representation system (i.e. Language): Which is why FOSS projects on the Internet are developed in a multitude of languages.

- The SFA system has been tried and tested mainly in my own practice. In India, it is possible to get into the details of each and every minute portion of a design. Whereas; I believe, the West uses a lot of prefab components. The uses of the SFA system in prefab components have not been tested yet.

Acknowledgments

Prof. Amit Bhaya; PEE/COPPE/UFRJ, Brazil and Kartikeya Date, MS Candidate

References

Frame, Michael; Mandelbrot, Benoît; and Neger, Nial: 2002; Fractal Geometry.
Francis, Sabu: 1987-2007: Works of architecture done at Navi Mumbai where the architecture designed was represented using the SFA representation system. See author's website.
Francis, Sabu: 1999: The importance of being abstract, eCAADe proceedings; Liverpool.
Kalay, Yehuda E: Khemlanai, Lachmi, Jin Woh, Choh: An integrated model to support collaborative multidisciplinary design of buildings.
Mitchell: 1994, Collaborative distributed design methodology.
Prusinkiewicz, Przemyslaw; Lindenmayer, Aristid: 1990; An Algorithmic Beauty of Plants (Springer-Verlag).