AN EVOLUTIVE ARCHITECTURE REASONS AS A SEMIOTIC, HERMENEUTIC AND AUTOPOIETIC ENTITY

Albertina Lourenci
LSI- Polytechnic School –USP
Av. Prof. Luciano Gualberto, 158 Trav. 3
Sao Paulo-Brazil CEP: 05508-010
Phone: ++55113091 5254
lourenci@lsi.usp.br

ABSTRACT

To build evolutive architectural and implementation code levels isomorphically to a domain level that generates sustainable cities autopoietically is challenging due to: 1) the likening of software to art is still in its infancy[14], 2) the highly interactive nature of the cooperative game exhibited by the experts has just begun to emerge [2] and 3) the needs of higher order reasoning structures mimicking the most evolved human cognitive processes such as reflective abilities, semiotic, hermeneutic and autopoietic reasoning require twining together many bits of fiber from diversified domains in order to see new formations and configurations of possible meaning.

Categories and Subject Descriptors

Software architecture: Composition and separation of concerns. Design patterns.

General Terms

Design, languages, theory, reflection.

Keywords

Domain, architecture, objects, semiotic, hermeneutic, autopoietic

1. INTRODUCTION

To build an evolutive software system means to unravel something that happens or that we want to happen in the world as information insofar this is aligned as an "evolutive white box" accessible to a wide gamut of intelligent systems with manifold cognitive abilities in the broadest contexts such as humans, molecules and machines. Inspiration can be drawn simply from nature and culture. Indeed life and language are information-carrying and goal-directed systems with intrinsic double articulation: the ability to create out of finite means something seemingly infinite.

The simplest although expressive form of evolution that strikes our eyes is metamorphosis in animals. The dicothomy between man thoughtlike and thing rocklike hinders us to see reality in a more meaningful way and as it really is.

On the one hand, Peirce (1839-1914) insists on: The thinking is not necessarily connected to a brain. It appears in the work of bees, of crystals…In the things around us one cannot deny it is there exactly like colours and forms are in them. If we cannot even dive into the meaning of a thing, the chance of unfolding evolutive systems seems remote [6].

It seems the object-oriented community has been thriving on this central idea of the thing striving to unfold its ultimate reasoning with varying degrees of success. Indeed the software community finally realizes what they build are intellectual artifacts. It is taken for granted that these "speak for themselves". Papers based on them seem unable to reach a consensus enabling computer science to help mankind to tame very large or highly complex problems such as the fact that today everyone needs to be a programmer. We need cheap, friendly, interactive languages and machines to communicate with people locally and all over the world. This shows a lack of basic understanding of the underpinnings of art.

On the other hand, until now people as non-linear, first order components in software development have not been taken into account [1]. Alistair Cockburn courageously highlights how challenging it is to understand experts through the conclusions drawn by Peter Naur (one of the authors of the BNT notation) about programming regarded as an activity by which programmers form or achieve a certain insight or theory on the matters at hand. After having examined real cases of the experience of evolving large systems, the latter perceived within the context of his Theory Building View:

1) The sort of theory advanced by programmers applies to constructions of specialized fields of enquiry as well as to activities that any educated person will join ocasionally

2) The dependence of a theory on a grasp of certain kinds of similarity between situations and events of the real world explains why it cannot be expressed in terms of rules or criteria. Likewise the similarities of many other kinds of objects such as human faces, tunes or tastes of wine cannot be thus expressed.

3) Moreover this theory remains a matter of the programmer's direct knowledge.

4) To modify a program means to recognize a similarity between the capabilities of the existing solution and the new demands. So only the human beings who possess the theory of the program in their heads can bring to light the design of the change of the program.

5) So the building of the program is the same as the building of the theory of it by and in the team of programmers

6) New generations of programmers of the theory of the program need to work in close contact with the programmers who possess the theory. This likens programming to artistic activities.

7) When this cannot be done, it is better to discard the existing program and the new-formed programmer team should be given the opportunity to solve the given problem afresh. It is more likely

to produce a viable program than program revival and possibly at a lower cost [2].

Hence both the nature of the cognitive processes of computer scientists and the object of their research have been likened to the nature of art. The challenge then is to understand art.

2. THE NATURE OF ART

The main hurdle to penetrate this realm is to be beyond cultural factors that compel humans to enter into a state known as subject-object dualism. So humans are not radically different from or better than other beings; but instead are moments in the play of phenomena. Heidegger and Buddhism claim that humans can learn to “let beings be” only by gaining insight into the nothingness that pervades all things. Such insight spontaneously leads to the overcoming of anthropocentrism and dualism. Gradually the mind perceives the world as a moment by moment manifestation of interrelated phenomena. All beings arise together simultaneously and are radically codependent in the sense of mutually defining one another [3].

The way the artist works also enhances this. To respect the material means to hear what the material wants to be by itself, let him tell you what to do. Brancusi tunes in: it is while you sculpt….that you uncover the spirit of the material and its peculiar properties. Your HAND THINKS and follow the material’s thinking.

This penetration and liberation of the immanent anatomy of the material follows another sort of emancipation, mining and liberation of raw material in the human unconsciousness, opening the gate for a spiritual dimension where egoism vanishes, man becomes metaphorically an open eye that sees everything especially the currents of the Universal Being circulating through him [4].

Moreover in this context it is easy to perceive both science and art aims at revealing truth. The difference is in their way of being. The gate for transdisciplinarity is open. Apparently it was closed by the founders of the Western thinking Plato and Aristotle. Friedrich Solmsen puts forward: in Physics, Aristotle deals with conditions and characteristics of physical objects without constrasting them with the properties of things eternal. The world of Forms is no longer as in the Timaeus (Plato) the background from which the happenings in the physical realm are set off. The discussion of place does not need realities that are not in place as a foil for those that are.

What’s the basic difficulty associated with this reasoning? Not only did it lead to departmentalization but also to attitudes such as to study what a mechanism can do, instead of what it was designed to do. So this controversial hinders humans to understand the true nature of the being of a thing while immanent and transcendent aspects.

Richard Gabriel proposed a deconstruction exercise at the Feyerabend Workshop at the Sixth European Conference on Pattern Languages of Programs held at Irsee Germany from July 4-8 2001. The goals of the brainstorm were to uncover hidden assumptions and devise new ways to move forward. The participants were invited to think of dozens of pairs such as science/art, aligned/contradictory, independent/crosscutting, subject/object, reflective/opaque to try to articulate what is wrong with the status quo of software and find new solutions [5]

2.1 The semiotic and hermeneutic value

As we saw above the world of our experience is the world of meaning. Heidegger argues that the goal of art is to reveal the truth in the form of a Gestalt (image, thing, geometric figure, artifact, model, etc). The being emerges during creation. Its unfolding appears as Poem. Hence all art is essentially a Poem in the widest sense of the word. He realizes that in the process of discovering things, there is a precise relation among the essence of the thing, the essence of the word and the truth. From the viewpoint of computer modeling, we cannot help delving deeper into this.

Semiotics focuses this better. Hence the world is also a domain of signs and a thing is also a sign. So we should stimulate ourselves to see a thing as open to inquisition. The world is a world of signs only because some organism interpret it. Signs do not exist independently of such acts. Likewise the thinking is also a sign and since life is a stream of thinking, a man is also a sign or thinking! The dialogue or a conversation is an obvious example of how the sign or the semiotic activity works. At any moment, signs are uttered by someone and interpreted by another. The semiosis or the action of the sign happens between two parts. Not necessarily people.

Peirce says the sign can be passive in relation to the object. Or active when it is the sign which generates the object. Hence a sign or the object of semiosis is both immanent and transcendent. While immanent goal, it must be identified with the Immediate Object, while transcendent being with the power and or force that constrains the sign in some way insofar it identifies itself with the Dynamic Object. All signs have Immediate and Dynamic Objects. Peirce says the Dynamic Object is a transcendent being that is alien to the fact it may be representable. There are three modes of being: possibility, reality and disposition (could be, real or would be). This is his general theory of the sign.

It enables us to model a thing, a being as a successive multitude of inner and outer parts of different degrees of preciseness and at different degrees of granularity as well as of different degrees of interaction ranging from autonomy to recursiveness.

We can distinguish the semiotic level (the level of intrinsically or inherently significant phenomena) and the hermeneutic level (the level of natural beings responding meaningfully to the things met in the empirical world)[6].

The structure of a language conditions the reasoning processes. Hence radical changes in reasoning cause structural changes in language. So modern semiotics extends the notion that anything is a sign to any phenomenon can be considered a semiotic phenomenon if it is simply analyzable as a function of a biplanar system or as a function of a monoplanar system or any system somehow structured and his characteristics have not been evidenced yet or even foreseeable. Hjelmslev’s language theory enables modeling the Immediate Object of the sign. Or its scientific aspects[7].

At this point, the artist is already thoroughly involved in this game. From now on to play simply fulfills its goal unraveling its own relationship with what is serious. The way of being of the play does not allow manipulation. This is the core where the way of being of the play becomes meaningful. Again like the semiotic value of the thing that is unraveled through an intelligent system and exists independent of interpretation, the game has its own nature and is independent of the consciousness of those that play. The game only exists when you respect the components of the game as equal partners independent of being things or humans. The thing must speak in equal terms. When one is able of this objectivity, it is easy to deal with subjectivity objectively! And respect it!

The movement of the game has no goal, but renews itself continuously. The way of being of the game does not demand a "subject" that behaves according to the template of the game to allow it to be played. Any one that knows how to play can start playing at any moment. The structuration of the game allows the gambler to bloom in oneself without any endeavour, making one feel pure joyness. Gadamer calls this an hermeneutic method.

Alistair Cockburn summarizes this wisely in his understanding of software development as a cooperative game of discovery, invention and communication [2].

2.2 An autopoietic domain model

Maturana and Varela define autopoiesis indirectly through how an autopoietic machine/system operates. As a characteristic process of self-production or homeostatic organization. It is the outcome of a direct trial to characterize the organization that makes living beings selfcontained unities and makes explicit the relationships among its components that must remain invariant under a continuous structural transformation and material processing [4].

To develop a domain model that reasons on the urban ecosystem as a semiotic, hermeneutic and autopoietic entity of course I underwent the cognitive processes described above. More about them and the ecodesign model entitled the Model of Primary, Secondary and Tertiary Waves (**MPSTW**) can be found in [7].

An architectonic object is its basic unit. It defines the urban ecosystem and is defined by it. A single construct generates it mimicking the nature of the DNA molecule and being applied to each element of the architectonic object, namely the activities, the structural system, the thermal comfort, acoustics, lighting, ecohydraulic installations, building system and so on. Its structure consists of the primary waves that displays processes to describe the interaction of the architectonic object with the environment namely homeostasis, continuity, differentiation and repeatability and the secondary waves responsible for the design processes, namely the substance of the function, the form of the function, the substance of the form and the form of the form.Thanks to it the urban ecosystem unfolds as a single organism, an autopoietic entity that is distributed in time and space by recursive partitioning into parts that are conceived similarly structurally speaking to tune in within the whole: Mother Earth, ecocontinents, bio-regions, ecocities, econeighbourhoods, ecobuildings, etc. However its arbitrary nature of the objects allows not only customization as well as decentralization. Any process or client or expert can enter this game at any point. And yet it is a true creative game.

The structure of the primary and secondary waves were determined first through the application of catastrophe, graph and semiotics theory specially Hjelmslev's language theory to shape the architectonic sign. It corresponds to the Immediate Object of the Architectonic Sign. Its underlying geometric modeling reflects the domain isomorphically. It is inspired by the tilings of the graphic work of M.C. Escher and mathematically unfolds as specially concerned with prototiles of different shapes, symmetry groups of the plane and the dotless plane. Similarity and conform symmetry groups are fractals [8]. The transcendent aspects correspond to the Dynamic Object that encompasses obviously the Immediate Object. The former is responsible for the theory underlying the development of the **MPSTW** . Although both aspects happen simultaneously, to model a theory of an ecosystem is a challenge not solved by the ecologists. Hence the evolutive nature of the model. It can be added to the model when possible.

3. THE HYBRID NATURE OF AN EVOLUTIVE ARCHITECTURE

The structure of an architecture behaves like a Janus face. It looks back towards the domain model and forward towards the implementation code obviously represented by a programming language. Hence it has a hybrid nature. If the transformation from the domain model to the architecture model and to the implementation model is isomorphic, the researcher has faced the crucible of integrating evolutive systems successfully. In this case a true metamorphosis is reached like catterpilar to chrysalis to "virtual butterfly" to butterfly. They obviously display a different nature because the models belong to different contexts in their path from reality to executing code. However the structure of the models must be isomorphic, if one wants to deal with evolution effectively and cheaply. In this case, both the architecture and the programming language must not only map core concepts from the MPSTW respectively but also display a friendly, interactive game structure.

Discovery costs are significant for evolving systems. They include both the costs of understanding the problem to be solved and the cost of understanding the existing software architectures and programming languages available. The highly abstract and sophisticated nature of the **MPSTW** as well its underlying geometric modeling not only allows the neat separation and composition of concerns as well as the transformation of each architectonic object both in its inner and outer configuration conforming to the neighbour architectonic objects in terms of function and form.

3.1 The basic level of the object

What are the key steps to implement the **MPSTW** ? Of course abstraction is the process of identifying similarities and then capturing them in abstraction mechanisms. There are two distinct aspects here: the discovery process and the mechanism. In previous sections, the discovery process was highlighted. Abstraction mechanisms, language features or constructs are the raw material of the low level of an architecture. Due to the highly arbitrary and dialogical nature of the **MPSTW** the object, class and inheritance mechanisms of the class-based languages are too coarse to implement it. The dialogical arbitrary nature is better grasped through prototypes, pure objects and the inheritance mechanism known as delegation represented by parent objects (traits) such as displayed by Self. Moreover Kansas, a multi-user programmable virtual reality allows cooperative work at a local or global distance depending on the bandwidth of high speed networks. Moreover its powerful GUI enhances the exploration of the thingness of the objects as well as the building of powerful graphical editors that may mimic free-hand sketch! Needless to say essential stage in the architectural design. It superbly presents exploratory programming [8].

3.2 The sublevel of views and layers

Yet the basic unit of the **MPSTW** is the complex green architectonic object. It displays views called waves responsible for the vertical object division and layers called the elements of the architectonic object responsible for the horizontal object division. The waves present a still finer granularity called the processes of the interaction of the architectonic object with the environment and the design processes. In natural language the elements correspond to the figures of the sign (prefix + radical +

suffix) and the processes to the letters. Of course the design processes are the vowels. These are the inner finer levels. Fortunately the so called subjective version of Self called Us [9] presents perspectives (views) and layers that correspond perfectly well to these finer levels. Moreover they are implemented as layer objects, perspective objects mimicking the game of art. Of course to date this has been the object of intensive research in the OO community.

3.3 The reflective metalevel and design patterns

But where are the higher level lingual abstraction mechanisms – those that specify the simultaneous interaction of several objects, a framework, a component? They are too costly to articulate as part of a programming language and too specialized to justify the price. Design patterns claim to enhance the power of **OO** mechanisms by capturing nontrivial relationships and interoperations. They may work as components and conectors [10]. One can build through them design pattern oriented frameworks or languages for architectural description.

Moreover how can one describe the various aspects of a domain where all reflective discussions exist only in terms of collections of these fundamental architectonic objects that autopoietically generate neighbourhoods, ecocities, bio-regions etc?

This demands reflective architectures. A reflective system (**RS**) is a causally connected meta-system that has as object-system itself. So it prescribes besides the computation about a domain, also reflective computation about itself. The metalevel (reflective level) reflects on the basic object level.

A higher order reasoning structure that has not been explored yet in **RSs** is a language's expressive power to describe itself. BNF, a formal notation to describe the context-free syntax of a given language can describe its own grammar. I intend to build an architectural descriptive language (**ADL**) based on design patterns reflecting isomorphically the **MPSTW**'s structure. Hence can design patterns describe their own design?

David Lorenz believes the Interpreter and the Visitor patterns of the GOF can describe their own design [11]. This is a fundamental reflexive relationship in pattern relationships.

Of particular interest to evolve the **MPSTW** is the process of assembling patterns by other patterns called pattern tiling. Lorenz is concerned with tilings of nonphysical objects as a metaphor to the true tiles that are made of stone or ceramic that cover surfaces without gaps or overlaps shaping tilings. Curiously the plan of an architectural and urban design maps to the concept of tiling nicely. At the level of the **MPSTW**, the tiles to be assembled are obviously nonphysical objects despite their geometric nature. Yet the final configuration is in fact a true tiling.

Hence the challenge is to build this reflective level within the context of a homo-iconic system built out of a single type of construct autopoietically [12]. Research to create a reflective Self is currently being accomplished [13].

4. CONCLUSIONS

To build a full-fledged evolutive knowledge system to generate sustainable cities that maps isomorphically its well advanced **MPSTW** concerned with immanent and transcendent aspects into the architectural and code implementation levels within the context of the **OO** paradigm is rather challenging at its current state of the art. Yet mankind must overcome this hurdle not only in order to build a sustainable basis for their cities but also to face the challenge of the many problems posed by mankind.

5. ACKNOWLEDGMENTS

Heartfelt thanks go to the anonymous referees, Roland Mittermeier, Mikio Aoyama, Jecel Mattos de Assumpção, Alistair Cockburn, Brian Rice, Randall Smith, James Coplien, Tom Mens, Galal Hassan Galal, Christian Wege, Pattie Maes, David Lorenz, Frank Buschman, Mehmet Aksit, David Ungar, Norai Romeu Rocco, João Antonio Zuffo, João Navarro, Gustavo Cerezo, Paulo Bressan, Cícero da Conceição, Jaqueline Albuquerque, Eliane Barbosa and Foundation of Protection to Research of São Paulo State for enabling me to express my ideas in the way of being of art.

5. REFERENCES

[1] Cockburn, A.: Characterizing people as non-linear, first-order components in software development. http://www.CrystalMethodologies.org/ 2001

[2] Cockburn, A.: Agile software development. Addison Wesley 2002.http://members.aol.com/humansandt/crystal/game/getasddraft.htm

[3] Zimmerman, M.E.: Heidegger, Buddhism and deep ecology in The Cambridge Companion to Heidegger edited by C. B. Guignon Cambridge University Press. 1999pp.240-269

[4] Lourenci, A.: New computational paradigms reflect the nature of an ecodesign model. The unfolding and modeling of artistic consciousness in the infoera. Postdoctorate Scientific Report I. FAPESP. October 2000. http://www.lsi.usp.br/~lourenci

[5] Gabriel, R.: FeyerAbend Workshop. EuroPLoP2001. http://www.dreamsongs.com/

[6] Colapietro, V.: Is Peirce's general theory of sign really general? Transactions of Charles Saunders Peirce Society. Spring 1987

[7] Lourenci, A.: Spirit, Energy and Information: Essential Elements of the Urban Ecosystem. Master's dissertation. Department of Architecture and Planning. EESC – USP. 1988. Summary in English available.

[8] Lourenci, A.: A proposal of a prototype based object oriented knowledge system to design and plan sustainable cities. PHD Thesis in Architecture and Urbanism. Faculty of Architecture and Urbanism. USP. 1998

[9] Smith, R. and Ungar, D.: A simple and unifying approach to subjective objects in Theory and Practice of object systems Vol.2(3), 161-178, 1996 http:// www.sun.com/research/self

[10] Bäumer, D.: Software-architecturen für die rahmenwerkbasierte Konstruktion grosser Anwendungssysteme. Informatik der Universität Hamburg. PHD Thesis. 2000

[11] Lorenz, D.: Tiling design patterns. A case study using the Interpreter Pattern. Proceedings from OOPSLA'97, 206-217

[12] Rice, B.: The Arrow system. http://tunes.org/ search for Arrow. 2001

[13] Assumpção, Jecel M. de: Self/R. http://www.merlintec.com:8080/software http://groups.yahoo.com/group/self-interest/links

[14] Coplien, J.O.: It's time to kill software engineering. Keynote at XV SBES Simpósio Brasileiro de Engenharia de Software –3/5 outubro, 2001 Rio de Janeiro Brazil.