Collective Design Network:

Systems Thinking (Event-Pattern-Structures) and System Dynamics Modelling as a Design Concept and Strategy

Nilüfer Kozikoğlu[1], Meral Erdoğan[2], Ahmet Kutsi Nircan[3], Fulya Özsel Akipek[4]
[1]Tuşpa NK, Turkey, [2,4]Yıldız Technical University, Turkey,
[3]Boğaziçi University, Turkey
[1]nkozik@gmail.com, [2]merale2007@gmail.com,
[3]aknxy@yahoo.com, [4]fulyaozsel@hotmail.com

Abstract: *This paper will relay the initial phase of a collaborative work within partners from the design discipline, systems engineering, and software engineering which deals with the interrelations of network idea, systems thinking, collective design, and computation. Vensim– a system dynamics modelling tool developed by Ventana Systems, Inc. in 1992 – has been used in an experimental first year design studio to engage students in systems thinking in the architectural design environment. It has been observed that this tool enabled most students to develop a multi-layered, complex and more controlled design logic and to amplify the cognitive processes at the beginning of the design education. We conclude that in order to fully realize systems thinking in the design process, new ways of integrating parametric design environments and system dynamic modelling environments needs to be investigated.*

Keywords: *Design network; system dynamics; dynamic pattern; collectivity; integration.*

Motivation

The need to set up collective design network in design process:

Architectural design process cover issues of the physical (light, visibility, movement, sound etc.) social (security, privacy, intimacy etc.) structural-material, and programmatic inputs and feed from various data sets and collective data sources produced and shared by interdisciplinary work and also operate within a network of multi layered information. Network approach generates the operations and methods to negotiate within these different data sets.

To develop a multi layered, complex and responsive design process, it is needed to externalize the interactions of multilayered factors and issues of design as networks and set a quantifiable cognitive model for the design intentions. Social behaviour patterns develop and change rapidly. Today social dynamics constitute more and more a design's subjectivities. This is the implication of the need of integrating the qualitative, intangible dynamic factors into the parametric design environment.

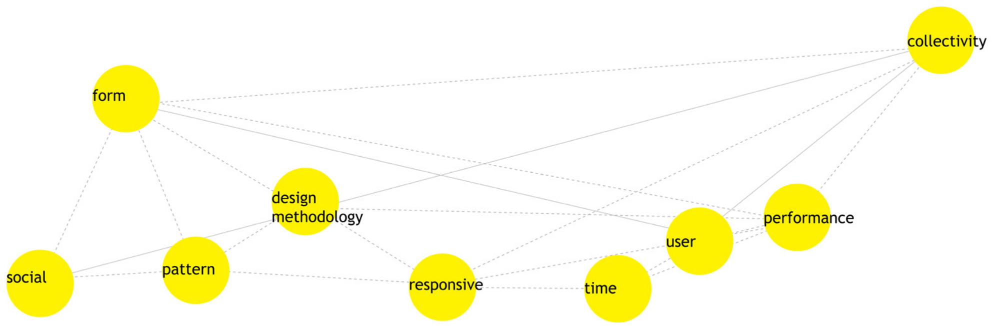


Figure 1
Keywords in writings on network thinking in architecture (diagram by Koseoglu in netwrokmetwork.blogspot.com: May 2009)

Network idea in design

Social and infrastructural contexts vary among multiple scales and interactions. Systemic and associative thinking employ scale-free design strategies and enable the incorporation of these individualities. Rajchman (2001) describes Deleuzian logic of senses as the logic in action in addressing multiple levels and trajectories inherent to contextual design activity. The mental, social and environmental ecologies (Guattari, 2000), and their inter-relativity require a logic that is not reductive, dialectic or transcendental, but one that is rather able to deal with multiplicities in an affluent manner and aims at invention and the discovery of potentialities.

In a top down or bottom up design strategy some of these individualities are eliminated during the reductive process. By logic of senses we refer to all the parameters of the design context that are collectively in tact, and live, either by feeding one another or at standby. Therefore the logic of senses is the network of parameters.

The number of research projects in between design field and network thinking has increased since 1990s with the acceleration in information technologies. Burke and Tierney (2007) provide an insight into network thinking active in the design realm. In their research on network practices the augmented relationship between connected environment and the space is investigated. Another source for network thinking in design is the collected works in AD journal edited by Hight and Perry (2006), Collective Intelligence in Design. A systematical scanning for the common topics and concepts mentioned in of design research projects marks the following concepts

We can denote that all projects we analyzed refer to concepts regarding collectivity and form, social, responsiveness and time, as well as pattern, design methodology, user and performance (Figure 2)

(http://www.networkmetwork.blogspot.com/: May 2009).

These notions of network thinking emphasize potentialities, a multiplicity in behaviors and interrelationship between parameters which hints at the complexity of design process. Computational tools feed this quality of multiplicity by enabling precision in the cause and effect relationships and the feedbacks.

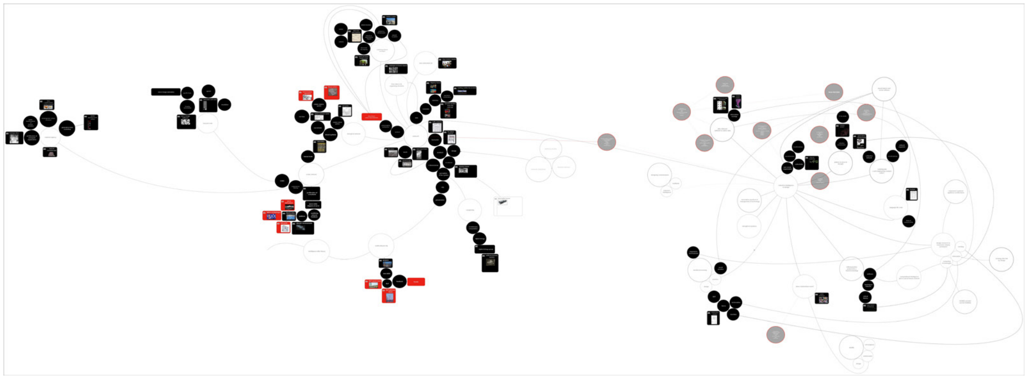
The way social, engineering and ecological studies deal with complexity in their fields is by systems thinking. In search of a methodology to engage in contextualization in complexity we referred in these disciplines.

Systems thinking

A system is a meaningful whole made of interacting parts.

One way of analysing real life systems is breaking them into parts or taking detached sections to find how the parts are working. In this reductionist approach parts gets more attention and it is believed that understanding how parts do work is key to

Figure 2
Mapping of concepts in design practices of network thinking. (diagram by Koseoglu in netwrokmetwork. blogspot.com: May 2009)



understanding how the over all systems is working.

Systems thinking provide a holistic approach to understand systems. Instead of looking at the parts, systems thinking looks at how the parts are interacting with each other and their environment to create the system behaviour. Systems thinking is useful especially for messy problems where the system is complex, and solutions with a reductionist approach create many unwanted, unaccounted side effects. Whether they are biological, social or ecological systems, many of the real life systems are complex systems which need a holistic approach. Design is also a messy problem which involves great complexity (Jackson, 2003).

Real life systems change over time and these changes are almost always not random. It is possible to see patterns in their behaviour. Complex systems may not show a very orderly behaviour but the variations in their behaviour are not random as well. Complex systems tend to show strange attractors where the systems behaviour varies around these attractors. Just like we cannot exactly know the weather in two weeks time but we know that it's hot in summer and cold in winter in the Northern hemisphere. Weather varies in between the pattern of seasons. The existence of strange attractors or patterns tells us that there is a structure governing the system behaviour. What the modeller tries to do is to predict this structure and the numeric values of the system to model the system at best. In short, systems thinking depict systems under a pyramidal organization where event has pattern that is immanent to structure.

Due to problems in validation and verification, it is impossible to correctly model a complex real life system. What the modeller can do is to come up with a simple enough model of the system which will mimic the main system behaviour that is under discussion. This simplification is different from the simplification of a reductionist approach. The outcome of the modelling in this case is still complex and non-linear and the model is formed from the point of view of the user. The system is simplified to bring forth the essential dynamics of the system. It is not simplified to be able to come up with an analytic solution.

System dynamics

System Dynamics (hereon referred to as SD) is a modelling technique introduced by Jay Forrester in 1950s (Sterman, 2000). SD was used to solve problems in social and physical systems especially in industry, urban development, world ecology, economics, etc.

SD simulates changes of the variables of a system over time. Ordinary Differential Equations (hereon referred to as ODE) are used to represent the real world system. With the advent of computers it became possible to solve any ODE numerically. SD uses computer programs to solve and display the changes of the model variables over time.

Forrester also introduced a diagram notation to depict models. This notation shows stocks or variables that are accumulating, as rectangles and flows (Figure 3) to and from these stocks as pipes with valves. Other variables that are not accumulating are shown without a rectangle shape and called auxiliary variables. Relationships between these variables are shown with arrows.

The idea of accumulation and flows is taken from ODE in calculus. SD and its notation makes ODE to be easier to understand and with the help of the software easy to solve.

SD models are usually complex enough to include one or more feedback loops. A feedback loop is a series of variables related to each other such that when the causal relationships are followed we encounter a loop.

Another model notation is also used to show the structure of a system. These models are called Causal Loop Diagrams (CLD). They can be considered as simplified SD models where most of the stock variables are omitted. CLD are easier to grasp but they cannot be simulated on computers. CLD are sometimes used at the initial stages of model building and understanding of the structure of the system under consideration. These diagrams consist of variable names and directed links between them. For example the causal relation between "weather temperature" and "number of picnic goers" would be represented with an arrow from the first variable to the latter with a positive sign on the tip of the arrow. The plus sign shows that as the "weather temperature" increases, the "number of picnic goers" also increases. A negative sign were to be used if this relationship was in opposite direction. As an example if the "number of picnic goers" increases, "available picnic space" decrease.

DYNAMO was the first SD software designed by Jack Puch at MIT (early 1960s) to enter the equations easily and then solve them numerically. Later software also included a canvas where it is possible to draw the model with icons of stocks, flows, auxiliary

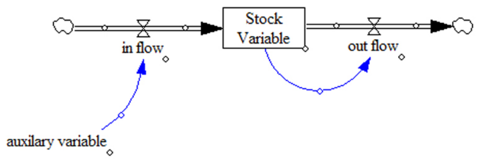


Figure 3
A diagram notation of system dynamics(Sterman, 2000)

variables, etc. We used Vensim for our work because it provides a free academic version and widely used among academic circles (http://www.vensim.com: May, 2009).

SD modelling allows intangible variables that are not possible or feasible to measure to be included in the model. Variables such as perception of a person can be used together with measurable variables such as the number of people. These intangible variables are quantified by the domain experts in the field and later can be tested if the system is sensitive to changes in these variables.

SD is a top down approach in modelling. Systems are modelled from the broader view and detailed as much as it is required. SD model is affected by the individual perception of the modeller. SD modeller holds an objective view of the real system and shall use real data, observations, and domain expert accounts as much as possible to create the model. SD modelling feeds from a collective effort.

SD requires experienced modellers with a good insight in the specific field obtained from domain experts. Dependence to experienced modellers is the weak point of SD approach. However, SD modelling is not a translation of the design intentions; it is itself a design process. Therefore collectivity and individuality characteristics of SD create a relevance of the use of SD modelling in architectural design.

Pattern language idea and dynamic design pattern

Christopher Alexander affected software engineering with his pattern language idea (Tignor and Myrtveit, 2000). The first application of pattern language was at the design level and was called design patterns. Today design patterns are widely used in programming. Later, patterns idea applied to analysis and requirements phases of software engineering as well.

The patterns that Alexander describes are similar to the patterns that are being observed in complex dynamic systems. They are, just like strange attractors, not exact behaviours but collections of similar behaviours. To use the same weather example, they are like the seasons describing general, abstract directions in behaviour (Alexander, 1979).

In software engineering, patterns, again, do not describe a single recipe that could be applied to all problems of design, analysis or requirements but, give a more abstract guidance. Both the software engineering patterns and Alexander's original patterns are attempts for dealing with complex, messy problems without exactly modelling the complete structure of the system under discussion.

Application

In order to arrive at a contextual and associative design platform we introduced SD to the first year design studio of The Faculty of Architecture in Yildiz Technical University in 2009. In this studio, network idea, collectivity and systems thinking were introduced to develop a multi layered, complex and more controlled point of view in the design process and amplify the cognitive processes.

As the transfer of pattern language to design in software engineering systems, we experimented in incorporating SD models in architectural design process. This is a re-appropriation of Alexander's ideas filtered through software and system engineering disciplines. Event, pattern and structure as the governing notions of systems thinking make up the major analogical shift into the design strategy.

We described the basics of CLD and SD with the help of Vensim. Students are encouraged to first decide on the time frame that they want to work with. For each case this time frame could be years, months, days or even hours. Then they are asked to think of fundamental variables and draw how they think these variables would change. Finally they tried to find loops in their projects.

We did not require students to quantify and run the model. The process of finding initial values of stock variables and writing equations is a long and difficult process for the novice. However running the model gives the real benefit of SD. When the SD model is simulated, it would be possible to see system behaviour for the given set of parameter values.

Students were asked to conceptualize an extension to an existing site at Anadolu Hisari, Istanbul, and design a temporary space or structure.

Figure 4 shows an example student SD model by Cansu Kilciler. The project was to build "temporary dog shelters." The rectangles show stock variables (variables that accumulate over time) while the large arrows represent flows to and from them. The small arrows represent the relationships between variables.

This model tells us that the number of stray dogs increases pollution and that, in return, affects the number of unhealthy people and the number of aggressive dogs. However as we can see the system does not have feedback loops that are necessary to reflect the complex dynamic nature of the system. We observed that many of the students found it difficult to find feedback loops for their projects.

On the other hand, once the feedback loops are understood, they give a valuable insight about the forces affecting the design. It is then possible to realize how everything in a particular design is connected to each other and how they change dramatically when the overall structure or an individual parameter is modified.

We also found that even naming of the system variables was a challenge. Students usually came up with vague naming for the variables. Making these more precise was an important process to better understand the system and dynamics of systems. As an example Nurdan Sezgin used "finishing of the working day" as a variable (Figure 5). In her model, when

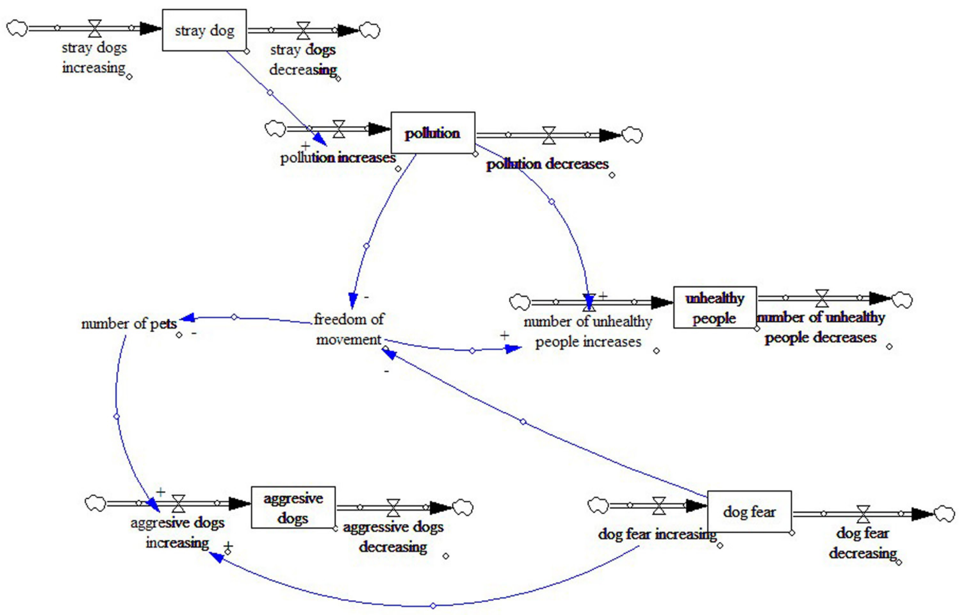


Figure 4
Cansu Kilciler's SD model via Vensim

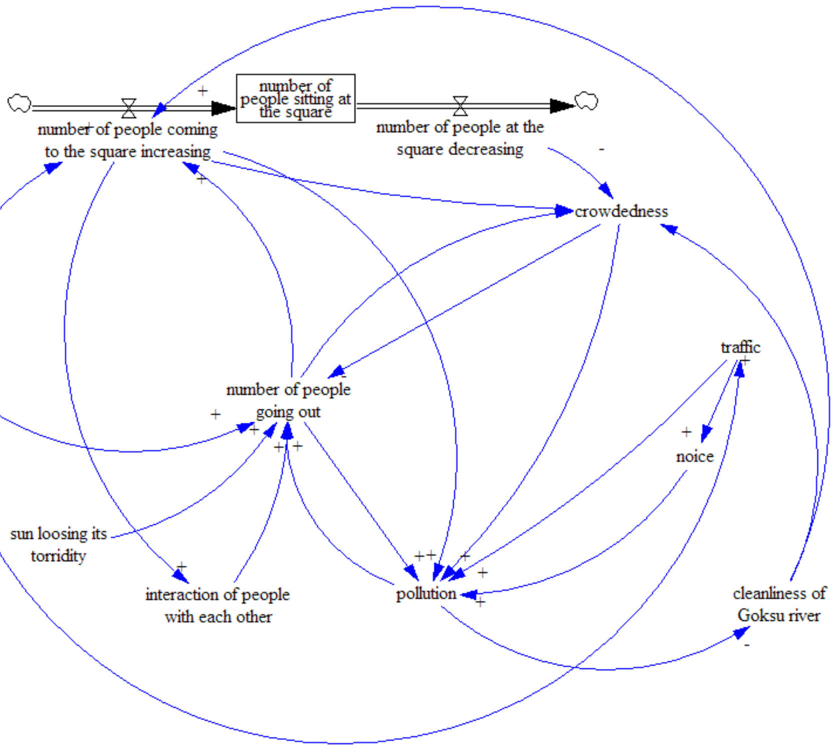
the working hour is over, you would see more people going out on the street but, "finishing of the working day" is one landmark on the time scale and it does not change its quantity over time. It is not possible to know its unit either. The systems thinking helped students to notice what really is affecting the systems behaviour.

Student SD models, drawings and other materials were shared with the studio using a social network site (Figure 6)(http://hisar-l-girisim.ning.com/, May 2009). It is possible to observe the development of individual SD models from this web site. We observed that students quickly understood the basic principles and started to ask questions about the systemic nature of their projects. They learned the difference between accumulating and non-accumulating variables and tested whether their causal links are valid.

Conclusion

Involving the network idea in the design process enabled the first year architectural design students to develop a mode of associative thinking about various layers of design issues related with the context of design and gain an attitude towards collective thinking. As seen in the examples of students' work in the application of system dynamics to design thinking, each of them identified the basic variables of their design intentions and the interrelations among these variables, before drawing a single line of their project. These variables were either concrete as i.e. "the number of stray dogs " or abstract as i.e. "dog fear". Collective thinking, in our point of view, entails an insight of a network of parameters intact, and an emphatic look at design from the points of views of various participants. It is also a platform

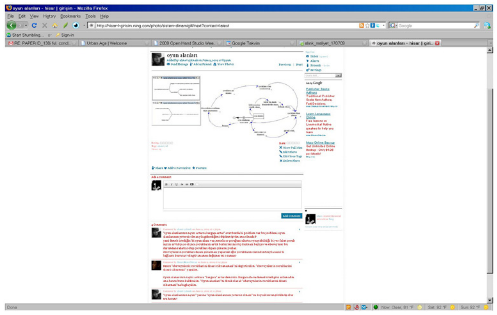
Figure 5
Nurdan Sezgin's SD model via Vensim


number of people sitting at the square
number of people coming to the square increasing
number of people at the square decreasing
crowdedness
finishing of the working day
number of people going out
traffic
noice
sun loosing its torridity
interaction of people with each other
pollution
cleanliness of Goksu river
+
-

for a collective design that is fed from various role players in the design process in an interdisciplinary manner.

System thinking helped the students to gain a holistic understanding in design issues. Rather than a reductionist approach to analyze the system by splitting it into parts, students tried to increase the relations within the different layers of design inputs. They inquired various data sets related to the given design subject, and the design site in the search for identifying existing and potential relations in the course of time, analysed the events in the search for patterns and aimed to reach a structure of related data sets that showed the way to develop a design strategy.

Figure 6
A sample view from the studio using a social network site (http://hisar-l-girisim.ning.com/)


oyun alanları - hisar | girişim - Mozilla Firefox
File Edit View History Bookmarks Tools Help
oyun alanları

The basic tool we introduced for systems thinking was one of the computational tools of system dynamics, Vensim. With Vensim, we had the opportunity to externalize the network of related data sets, involve the qualities as well as quantities of design inputs, compute them as equations, and identify the affects of these relations to each other as feed back loops within a course of time. With this tool, we test the performance of the design strategy of the student, i.e. the designer, and reach a more methodological approach via this externalized cognitive model because we had the chance to see all the variables and interrelations of the students' intentions of design before commenting on the spatial qualities of it.

Systems thinking and system dynamics helped us to introduce the notions of time, associative thinking, parametric design and multiplicity to first year design studio while the tool- at that instance Vensim, created some barriers in design research to shift the cognitive model to design operations. While the design strategies, the design program that the students offer and the design parameters which they reveal were excellent, but they had the difficulties to pass these ideas to material and spatial organization.

The unique motivation in devising this studio exercise and approach was to direct the students for collective design thinking which entails to think of design as a network design process, as a research for setting a system which generates multiplicities and potentialities and which is generated in the collectivity of the role players and all the related disciplines.

The first year students confronted the multilayered architectural design criteria through SD modelling. Building up a network of design components prevented them arriving at premature and cliché design production. All projects has integrity within the criteria it proposed and had strong references with the design site; mostly the community. SD models were operative in challenging students' individualities; however we witnessed the gap in translation of the network setup into dynamic form properties.

The applications of the students, and the barriers they faced to operate in this process showed the need of an interface that feeds in and out the SD model with the parametric design modelling. A computational design environment that enables the interactions and feed backs within the SD model and parametric design modelling in order to trace the design process and get the full advantage of a computational medium in design research is our further goal of research.

References

Alexander, C.: 1979, The Timeless Way of Building, Oxford University Press, New York.

Burke, A. and Tierney T. (ed.): 2007, Network Practices, Princeton Architectural Press, New York,pp.100-115.

Guattari, F.: 2000, The Three Ecologies, The Athlone Press, New York.

Hight, C. and Perry, J., (ed.): 2006, Collective Intelligence in Design, Architectural Design, 76 (5).

Jackson, C. M., 2003, Systems Thinking: Creative Holism for Managers, John Wiley & Sons Ltd., West Sussex..

Rajchman, J.: 2001, The Deleuze Connections, Chapter 4: "Multiplicities", MIT Press, Boston, pp. 49-76.

Sterman, J. D.: 2000, Business Dynamics : Systems Thinking And Modeling For A Complex World, Irvin/Mc Graw-Hill, Illinois.

Trummer P.: 2006, On Multiplicity: Population Thinking, The Berlage Institute Research Report (8), pp.69-75.