Towards Socially Embedded Software Engineering

Introducing Social UML

Ravi Shukla (CSSP, JNU, New Delhi)

Abstract: The typical software engineering process entails translating real-world social or human problems to a technical ones. The process of translation and the subsequent analysis, design and implementation steps tend to exclude social, political and cultural factors in favour of more obviously technical ones. These distinctions between the technical and the social may not always be well defined and the supposedly non-technical factors may play a critical role in the design and outcomes of the software. Building on the concepts of Cultural Historical Activity Theory (CHAT) and Interaction Design, this paper suggests ways to extend the existing UML vocabulary so that the complex sociotechnical process of software development includes more social aspects.

Keywords : Socially Embedded, UML, Social UML, Software Engineering, Activity Theory, CHAT, Interaction Design.

1. Overview

At the outset it should be clarified that 'social' aspects include political, economic, cultural, psychological and other factors that are generally seen as 'non-technical'.

'Socially embedded' in the title has been adopted in to the current context from the work of Karl Polanyi (1886-1964), a Hungarian economist. Polanyi spoke about the 'embeddedness' of the relationship between human society and the market. According to Polanyi, the most economies reflect the 'subordination of human society to the logic of the market', so that instead of the economy being 'embedded' in social relations, 'the running of society was an adjunct to the market'.[i] There are some tacit assumptions in the attempt to include more social elements in the development of software systems. First that there is *scope* for inclusion of more social aspects in the development of software systems and secondly that it is *desirable* to do so.

1.1 Technoscientific Systems in General

The inclination to include social aspects or considerations in science and engineering processes seems to be applicable to technoscientific systems in general. Here *technoscience* suggests the entwined nature of science and technology, indeed some are of the view that science is inherently a technological endevour and and the Kuhnian notion of scientific revolution may be better illustrated by a change in the technical apparatus rather than in the conceptual framework of science.[ii]The increasing acceptance of global warming coupled with disasters such as the Fukushima nuclear reactor in Japan seem to have led to realization that some rethinking needs to be done in how we approach the conceptualization and implementation of technical systems. [iii] [iv] The extensions to the software modeling approach may open the ground to similar inclusions in other fields.

In a poll conducted on the possible ways to address scientific and technological problems and issues, of all the possible answers, one suggesting a reconceptualizing of science and technology so that it is more in tune with social and environmental realities, proved to be the most popular. Put differently, the majority seemed to favour the inclusion of more social and environmental aspects in the conceptualization of science and technology.

A summary of the responses is shown in table 1.

Table- 1. Summary of Problems and Issues of Technoscience.

Option	Num[1]	%
There needs to be more involvement and awareness of social aspects without compromising on autonomy	5	14
Science and technology cannot be equated, science is about knowledge, its application(technology) may have harmful effects	4	11
These problems come about due to other social and political problems and the onus is on science and technology to solve them	1	3
There is a need to reconceptualize technoscience so that it is more in tune with social and environmental realities	22	61
It's like catching a tiger by the tail, once you catch it you can't let go…	1	3
There are unintended problems, but these can only be solved by more science and technology	1	3
None of the above	1	3
Other	1	3
Total	36	100

http://stasis.in/PollTechSciProbs

1.2 Social involvement in Software Systems

In the case of software systems, we start our attempts at social inclusion by attempting to include users the engineering process.

[1] Denotes the total number of votes, % denotes the percentage of the total for this opinion

As the response to the question below seem to suggest, most people in the software industry are of the opinion that more user involvement may lead to lower costs and better quality.

Question: Do you think involving users in analysis and design will a) increase the cost or b) reduce the reduce the technical quality?

Table-2: Effect of User Involvement in Analysis and Design.

Option	**Num**[2]	**%**
both a) and b) are true	1	6
both a) and b) are false	1	6
a) is true b) is false	3	18
a) is false b) is true	1	6
More user involvement may lead to lower costs and better quality	10	59
Don't know can't say	1	6
Total	17	

http://stasis.in/SoftwareUserInvolvementCost

In the typical software engineering process the interaction with users and user groups, is generally limited to the initial stages such as the requirements gathering phase or else later in the acceptance testing phase. Involving user voices in other stages such as analyis and design decisions is relatively lessor known. Attempts such as Joint Application Design (JAD) while they attempt to seek interaction between the users and the design team beyond the requirements phase, do not seem to have a mechanism to include the larger social context. [v]

The 'Social context' here tends to get tacitly represented in the background of categories, concepts, ideas and assumptions that often manifest in the metaphors used. Cognitive scientist George Lakoff is of the opinion that there is 'nothing more more basic than categorization to our thought, perception, action, and speech'. Most categorization, according to him, is implicit; it is automatic and unconscious and we are barely aware of it.[vi] George Basalla, who uses the metaphor of evolution, to describe the process of technological change as opposed to others who view it as the outcome of individual brilliance, suggests that metaphors are 'at the heart of all extended analytical and critical thought'.[vii] While there may not be any foolproof way to include these aspects in the otherwise 'technical' process, this paper attempts to outline a mechanism that may *enable* this inclusion. The sections that follow outline the basis and the mechanisms of these inclusions.

2. Activity Theory

Marx and Engel, building on the work of Hegel and others, suggested that the human narrative was more than 'the deeds of great men', a notion prevalent at the time, or an 'accumulation of accidents'. The central question, according to them, was the economics of the labour process.[viii] Activity theory, while acknowledging the importance of the labour process, seeks to extend the notion to other areas of human activity. It also attempts to see these activities in their cultural, historical context, lead to it being dubbed 'Cultural, Historical Activity Theory' (CHAT). The abstraction of the notion of activity, makes available a shared language and conceptual framework across diverse disciplines so that an activity would have the same meaning for a "political scientist, psychologist, political economist, neuroscientist or a sociologist". An 'activity' in Activity theory may be loosely described as 'a system of actions in pursuit of some socially determined object', or an object the motive for which resides in the requirements of society rather than the needs of an individual'.[ix]

2.1 Technology and Activity

Each activity involves a subject, or the performer of the activity, the object to be achieved and the tools used to achieve some specific outcome. Interestingly, tools, often considered a 'uniquely human' characteristic, are seem as mediators between 'internal' mental processes and the external world.[x] Indeed, activity theory suggests that that the tools used to perform an activity gradually become a part of 'internalized' 'mental' processes. [xi]In other words, our roles and tasks in complex socio-technical networks also tend to play a role in our cognitive processes. Other theories imply that collaborative activities such as flying an airplane or developing software is indicative of common goals and a shared understanding of abstract concepts.[xii] The relationship between subject, the object and the tools, has evolved over time to include a set of implicit and explicit rules that defined the boundaries within which the activities must be performed, the communities of people involved and division of labour or who is to do what. This is shown schematically in the diagram below:

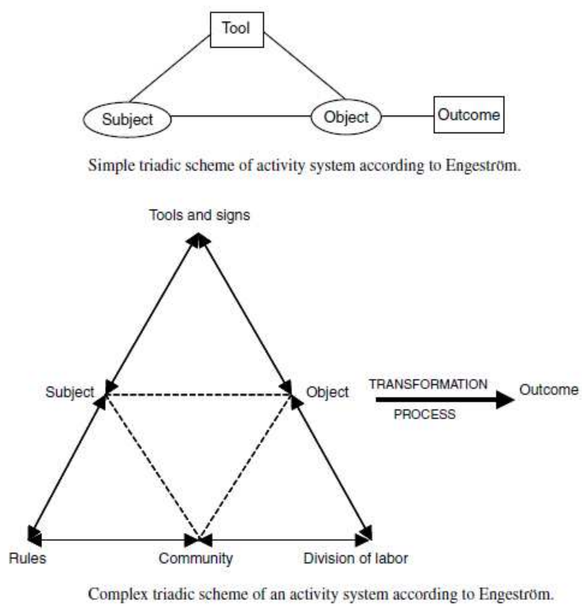


Fig 1: Activity Theory Schemes[xiii]

2.2 Some characteristics of Activity Theory

Authors, Keptelinin and Nardi find the activity theory framework useful for people negotiating the 'thickets of users

[2] Denotes the total number of votes, % denotes the percentage of the total for this opinion

and their needs, and technologies and their possibilities'. The authors outline some broad tenets that they find useful. These are:

- an emphasis on human intentionality;
- the asymmetry of people and things;
- the importance of human development; and
- the idea of culture and society as shaping human activity.

Intentionality, in this context suggests a certain deliberation or intent in performing certain acts with certain technologies. The intentionality extends to the technology process where, 'behind every design is an intention', just as there is an intention behind every action. People or subjects work with technologies towards accomplishing their intentions as objects or goals. This suggests that the relationship between tools and people is one of *mediation* in that tools mediate between people and the world. Asymmetry, or the difference between artifacts and human beings, the authors point out, is one of the distinctions that distinguishes this approach from others such as the Actor-Network Theory that posit a sociotechnical network whose 'nodes' can be either. *Development,* in the context of activity theory does not seem to suggest a linear, universalized notion applicable across cultures and technologies but one that brings a historical perspective to the activity under consideration. Development is seen as a dialogic process wherein people cycle, grow and change with technology and may also transform culture through their activity. The historical frame suggests that adoption of a technology is not sufficient criteria for success and wider impacts and longer term perspectives need to be taken into consideration – such as adequate means of handling byproducts or waste – for instance in medical or health related activity.[xiv]

3. Design Influences

As with activity theory in the previous section, a detailed discussion of design theory is outside the scope of this paper. At best, I will try to identify some aspects that may be relevant to the UML extensions to be described later. Let me build on the somewhat obvious notion that the awareness and imagination of the engineers and experts responsible are likely to have an ifluence in the conceptualization and design of the systems. This is brought forth by Michel Callon's classic study of the design of the VEL electric car in France in the 1970s. Callon noted that the designer's of the technology must not only find answers to engineering problems but also imagine the social universe in which the system must function. In this process they end up working as de-facto historians and sociologists.[xv] What this seems to suggest is that the assumptions, metaphors, concepts, categories and metaphors in the background of the problem-solution space, may influence the design and outcome of the system. So do the existing tools and artifacts; as has been noticed, in some cases, the availability of pre-existing technical sometimes leads to a tendency to redefine the problem in terms of the available solutions.[xvi] The relevance of metaphors is exemplified in the field of software by is seen in the change metaphors that marked the shift from a structured programming paradigm where software was seen as 'a sequence of steps' to the object-oriented paradigm where the commonly used metaphor was 'software as a collection of objects' where objects were the abstractions of real-world entities in the problem space. This also indicated a shift in the perception of the discipline 'computer programming to software engineering'. More recently there has been a trend towards designing software components as relatively autonomous 'agents' with pre-determined goals that may interact with other software components so that one may view software are 'a society of agents'.[xvii]

Equally important in technology development is the question of which of the myriad different voices is heard in the process of reaching an agreement on finalizing requirements, making design choices, deciding upon success criteria and so on. As Andrew Feenberg has pointed out, the process of decision making, masks the historicity of interests and the politics of the specific technology. Each such technology, argues Feenberg embodies some particular people's decisions/power over which among many possible considerations or voices are to be included and which are to be excluded in the technology. Feenberg suggests that the involvement of lay users and an active engagement with active questions of who determines technology choices, at what cost, to what end by what institutional or social process leads to a restoration of human agency.[xviii]

Another design problem that needs to be addressed is that more often than not, problems may not be clearly defined at the outset and generally become clearer over time. 'Well-structured' problems, or those in which it may be possible to acquire gain more information about the problem, identify the starting state, the 'solved state' and a set of intermediate states are relatively rare. To make the transition from an unstructured or semi-structured problem to a clearly defined one that includes a series of practical steps to solve the problem as well as a set of repeatable criteria to test the solution is one of the challenges of design.[xix]

4. Social UML Overview

At the outset of this , it is important to make a distinction between Social UML and Agent Oriented UML or other related UML extensions which concern themselves with 'social aspects' of Multi Agent Systems (MAS). Here the agents are software components and their *social structure* as the properties of the societies, groups and institutions that may exist, and the roles that these 'social' entities play in their relationships. *Social behavior* includes dynamics such as the temporality and causality of events, interactions such as those between individual agents and societies, a set of norms or rules that agents generally follow and a set social *attitudes* that indicate common tendencies – intentions, motivations and so on.[xx] Since the two could share similar terminologies, it is important to reflect the *asymmetry* between human and technical or non-human agencies in the engineering process.

Before proceeding with it should be noted that that the *spirit* of embedding technologies in their social context, rather the specifics in this proposal that can perhaps act as the beginning to a more comprehensive description.

The proposed extensions in this section assume UML version 2.0 to be the last major revision. At the time of writing the last revision has been 2.4.1, and does not include the extensions proposed here.[xxi] The proposed extensions to UML fall under three broad categories.

- Inclusion of Newer Diagram Types
- Inclusion of Newer Concepts

- Inclusion of Newer Attributes and Types

Of these the third and possibly some aspects of the second category could possibly be included in the notion of UML profiles.

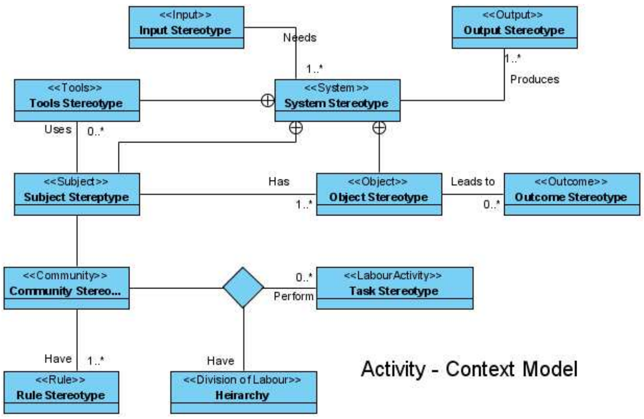
4.1 Meta Model of Extension Elements

The meta-model attempts to provide a common platform to place the techical system in a social context. This includes the diverse voices related to the sytem, the background of assumptions, concepts, metaphors and intent, the main activities and technical tools used and the criteria for success and failure.

4.2 Diagrams Proposed

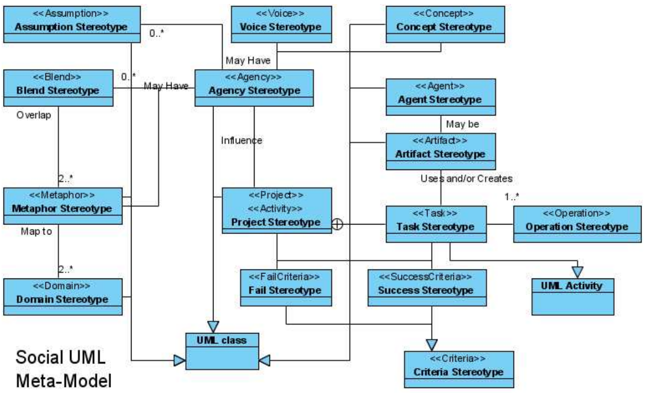
4.2.1 Activity-Context Diagram

The *context diagram* in traditional structured analysis and design giving an overview of the system as a whole by providing a one line description of what the system does and listing its inputs and outputs. *Activity theory* suggests the the idea of a subject using tools towards achieving specific objectives and resulting in certain outcomes. Also relevant in this picture are the rules, communities and divisions of labour in this activity. A blending of these two notions provides the template of an Activity-Context diagram. The model is shown below, an occurance of the diagram for a case study is shown in the next section.



4.2.2 Problematique Diagram

The problematique diagram provides a snapshot of the problem space by including the main assumptions, domains, concepts, metaphors, contraints, voices, agencies and tasks and the technical tools and artifacts. The model is shown below, an occurance of the diagram for a case study is shown in the next section.



4.3 Introducing Temporality

The notion of time is relevant both at the level of the system – where for instance, a missile control system needs to respond in real-time as opposed to a query about 'employees' in a organization - as well within the system, where some data elements may change more rapidly than other.

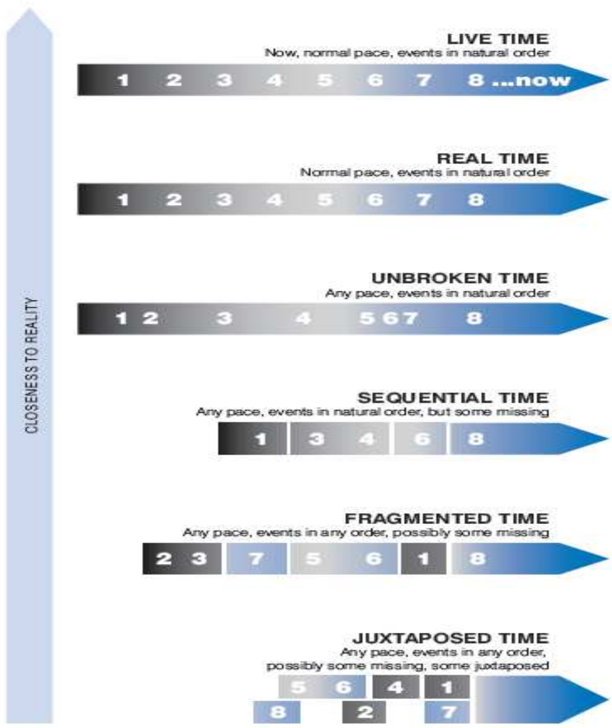
4.3.1 Periodicity

Traditionally, the rate at which different data elements change have been dealt with by classifying them as 'master' or relatively unchanging data such as item name, item identifier etc, and 'transaction' data such as num of items bought or sold, date and time of purchase etc. The difference in periodicity also has implications for the relationships between objects, for example if one of the related attributes of an object changes, it effects other related objects as well. These are made more explicit and by classifying and quantifying them in association with each other.

4.3.2 Types of Time

Depending on the nature the system under consideration, different notions of temporality may be relevant. To give an example in some systems, 'live time', or the time that we live by, may be more relevant for example for controlling intensity or light or brightness depending on the time of the day, or automatically adjusting the number of days in month depending on the day and year; in others, such as games or other simulation applications, 'real time' or the time elapsed may be more relevant. A diagrammatic representation of different notions of time is shown below:

Fig 1: Different notions of time



xxii

4.4 Additional Elements and Attributes (profiles)

In order to include notions of temporality, that the following elements and attributes may be added to existing elements in UML. While these are described here in brief, their usage may possibly be clearer at the end of the case study in the next section.

For 'class' and 'attribute' elements, the following attributes: Likelihood of change – this can be some simple literal – very likely, somewhat likely, little likely or unlikely

- Rate of change – several times in each application run, at least once per execution, daily, weekly, monthly.

In addition, for classes, the following aatributes may be included, their values would reflect the 'real world' entity that the class represents.

- Expected Life Span – this reflects the 'real world' representation, for example for class 'employee', the 'expected life span' could mean the retirement age in the organization.
- Archival Procedure – What is the procedure to be followed when the expected life span ends.
- Exit Procedure – What the is the procude to be followed for changing the status of the object.

Most of the elements used in the Activity-Context Diagram have been described in the earlier sections. We now turn to the description of the elements of the problematique diagram.

4.4.1 Problematique Elements

Project: Most UML tools (Visio and Visual Paradigm to name two) have the notion of a project, however it is generally not included as a UML element. In addition to other attribute generally present such as start date, end date etc, some new attributes may include initiating agency (Govt/NGO/Public Service/Corporate/SME) and so on; project type (product/service/or both); project domains (required areas of expertise), Temporality (live time, real time, unbroken time, sequential time, fragmented time, juxtaposed time and so on), motivation (altruistic, profit, entertainment, support) and so on.

Task: each task is a part of the project, a task can consist of a number of operations performed in a particlar sequence. Each task is initiated or performed by an actor, has a set of inputs and outputs, a set of expected outcomes and may have success or failure criteria.

Domain: describes the area of expertise or knowledge which is required for the project. There could be multiple domains, each with a description, number of experts required, time for which they are required and so on.

Metaphor: a metaphor is generally in the background and maps to a domain. It may or may not be connected with some concepts, contraints or assumptions.

Assumption: an assumption is generally connected with one or more of the elements in the project.

Constraint:a constraint generally sets some limits on one or more project elements.

Agency: 'agency' is generally attributed to entities that have the capability of influencing their own social environment. As mentioned earlier, here the social includes political, cultural, environmental and economic factors.

Artifact: artifacts are generally tools that are used in the project, these could be technical tools such as hardware and software or conceptual tools.

Voices: each project has a diversity of voices, these include the voices of the users, project initiators, domain experts as well as others not directly involved in the project development.

5. Case Study

The case study is that of Hospital Information System of a privately owned hospital. The subjects here are the hospital management or the Board of Directors of the hospital. The objective is to make a profit by providing health services. The various services offered include diagnostics and treatement for with the associated billing and support actions tasks. These are performed by Doctors, Nurses, and other Hospital Staff who use a diverse variety of technical tools to help them in their tasks. Each that tasks can comprise a number of sub-tasks or operations. This brief overview is represented in the following diagrams that are representative of the system. These are not complete representations but are only presented here to provide an idea of usage

Fig Hospital Activity-Context Diagram

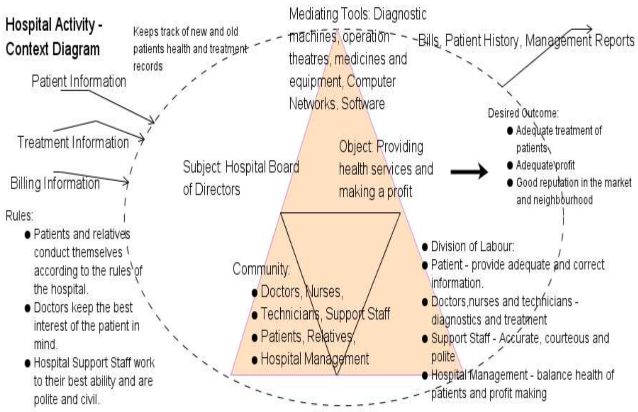
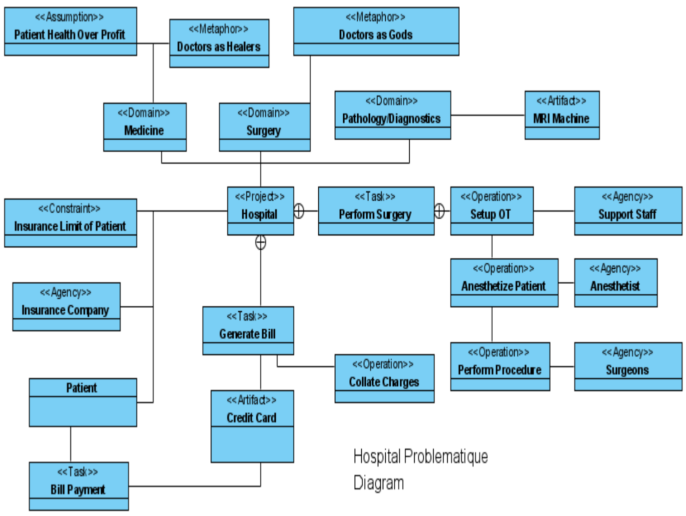


Fig: Hospital Problematique Diagram



(Manuscript received Sep. 25, 2002, revised May 12, 2003)

References

i Kaan Agartan, Kayse Bugra, *Reading Karl Polayni for the Twenty – First Century: Market Economy as a Political Project*, Palgrave Macmillan, New York, Ny, 2007. pp 1-3.

ii Mitcham Carl, *Thinking Through Technology: The Path between Engineering and Philosophy*, University of Chicago Press, Chicago, London, 1994. pp 97-99:

iii Nisbet, Matthew C. and Myers, Teresa, 'The Polls—Trends: Twenty Years of Public Opinion About GlobalWarming'. *Public Opinion Quarterly*, Vol. 71, Issue 3, pp. 444-470, 2007

iv The Guardian, Japanese energy policy stands at crossroads, Thursday May 3, 2012. Last Accessed 06 June 2012 14:00 IST

http://www.guardian.co.uk/environment/2012/may/03/japan-nuclear-power-post-fukushima
[v] Yatco, Mei C. "Joint Application Design/Development." Management Information Systems Program. Fall 1999. University of Missouri-St. Louis. 22 June 2006. Last Accessed 09 June 2012 14:00 IST <http://www.umsl.edu/~sauter/analysis/JAD.html>
[vi] Lakoff George, *Women, Fire and Dangerous things: what categories reveal about the human mind,* University of Chicago Press, Chicago, 1987. page 6.
[vii] George Basalla, *The Evolution of technology*, Cambridge University Press, Cambridge, UK, 1998. page 3.
[viii] Fischer Ernst, *Marx in His Own Words*, Penguin Books, Middlesex, England. First Edition 1973, Reprinted 1984. Page 81.
[ix] Blunden Andy, *The Interdisciplinary Theory of Activity*, Brill Publishing, Leiden, Netherlands, 2010. pp 170-172.
[x] Bedny Gregory, Karwowski Waldemar, *A Systemic-Structural Theory of Activity: Applications to Human Performance and Design*, CRC Press, Taylor and Francis Group, Boca Raton, Fl, 2007. pp 4-5.
[xi] Kaptelinin Victor, Nardi Bonnie A, *Acting with Technology: Activity Theory and Interaction Design,* The MIT Press, Cambridge, Mass, London, England. 2006. pp 42-44.
[xii] *Ibid*, page 196.
[xiii] Bedny Gregory, Karwowski Waldemar, *A Systemic-Structural Theory of Activity: Applications to Human Performance and Design*, CRC Press, Taylor and Francis Group, Boca Raton, Fl, 2007. Page 39.
[xiv] Kaptelinin Victor, Nardi Bonnie A, *Acting with Technology: Activity Theory and Interaction Design,* The MIT Press, Cambridge, Mass, London, England. 2006. pp 7-13.
[xv]Callon Michel, ‘Society in the Making: The Study of Technology as a Tool for Sociological Analysis’, in Wiebe E Bijker, Thomas P Hughes, Trevor J Pinch Ed, *The Social Construction of Technological Systems,* The MIT Press, Cambridge, Mass. London, England. 1987 pp 84-85.
[xvi] Hughes Thomas P, *Networks of Power Electrification in Western Society(1880-1930)*, John Hopkins University Press, Baltimore and London 1983, page 286. Hughes gives the example of the large electric power generating stations built during the first world war to fulfill the unusual wartime needs. Once the war was over, these stations became a ‘solution in search of a problem’. And there was an effort to maintain in peacetime the ‘momentum of their wartime planning and nurturing of technology’.
[xvii] Imaz Manuel, Benyon David, *Designing with Blends: Conceptual Foundations of Human-Computer Interaction and Software Engineering,* The MIT Press, Cambridge, Mass, London, England, 2007. pp 12-13.
[xviii] Gerald Doppelt, ‘Democracy and Technology’, in Tyler Veak Ed, *Democratizing Technology: Andrew Feenberg’s Critical Theory of Technology,* State University of Albany Press, Albany, New York 2006. pp 85-87.
[xix] Kolko Jon, *Exposing the Magic of Design: A Practitioner’s Guide to the Methods and Theory of Synthesis*, Oxford University Press, Oxford, UK, New York, USA, 2011. pp 4-6.
[xx] Cervenka Radovan, Trencansky Ivan, Calisti Monique, ‘Modeling Social Aspects of Multi-Agent Systems: An AML Approach’, in Muller, Jorg P, Zambonelli Franco (Eds), *Agent-Oriented Software Engineering VI*, 6th International Workshop, AOSE, 2005. Lecture Notes in Computer Science 3950, Springer-Verlag, Hiedelberg, Berlin 2009. pp 28-29.
[xxi] Selic B, ‘Models, Software models and UML’, in Lavagno Luciano, Martin Grant, Selic Brian Edited, *UML for Real: Design of Embedded Real-time systems,* Kulwer Academic Publishers, New York, Boston, Dordrecht, London, Moscow, 2003. pp 8-10.
[xxii] Lundgren Sus, Hultberg Theo, ‘Time, Temporality and Interaction’, *Interactions,* July + August 2009, Association for Computing Machinery (ACM), pp 34-37.