Re-Cognizing Software Modelling: A SocioTechnical Approach

Ravi Shukla
Research Fellow, Centre for Culture
Media and Governance
Jamia Millia Islamia
New Delhi, India
+919811421221
ravishu@gmail.com

ABSTRACT
The mutually reinforcing relationship between information systems and cognitive processes are reflected in metaphors such as the notion of the mind as an information processing machine and efforts such as machine learning and neural networks. The predominant understanding of cognition - that functions such as memory, learning and reasoning involve the storage and manipulation of images and symbols also pervades software modelling and design - itself a cognitive activity. However, a somewhat different understanding of Cognition - as a dynamic and adaptive activity of responding to one's physical and social environment also points to newer approaches in modelling. This paper suggests possible extensions to the Unified Modelling Language (UML) that take on board domain specific and social aspects and activities and include them in design models.

Categories and Subject Descriptors
D.2.11 [Software Architectures]: Data Abstraction, Domain Specific Architectures.

I.2.4 [Knowledge Representation Formalisms and Methods]: Frames and Scripts, Representation Languages

General Terms
Design, Human Factors, Theory, Languages

Keywords
Software Modelling, UML, UML Profiling, Object-Oriented Design, Cognition, Cognitive Models, Activity Theory.

1. INFORMATION AND COGNITION
The binary nature of neuronal spikes as the underlying mechanism for transmission of signals in the brain [1] suggested a connection between brain function and digital computing almost from inception [2]. This may be traced in the relatively well known concepts such as neural networks and machine learning in the field of computer science and notions of cognitive architectures such as ACT-R (Adaptive Control of Thought-Rational) which attempts to create software that simulate and perform cognitive tasks.[3] Other efforts that take cognitive aspects on board are those of Computer Haptics, Ergonomics and Human Computer Interaction (HCI). What most of these approaches share is the idea of 'disembodied' cognition or the notion that the 'body' of a real-world object may be dissociated from its image or symbol representation in the cognitive system. In other words, they assume a pre-existing world which may be represented as symbols or images in the cognitive system that may then be stored, recovered and manipulated.[4] Information systems too, have traditionally seen the information about a body as separate from the body itself, so that the "flow" of information between the constituent parts is essential to the conceptualisation of the body.[5] More recent theories that take cognitive aspects on board have tended to focus on the meaning of information and the relation between its elements rather than viewing it as an objective body independent of relations.[6]

A more embodied notion of cognition has been seen by neuroscientist Francisco Varela and his co-authors Evan Thompson and Eleanor Rosch as one that depends on the experiences that come from having a body with sensorimotor capabilities which are themselves embedded in a more encompassing biological, psychological, and cultural context. They also see perception and action as fundamentally inseparable and co-evolving in what they call, "lived cognition".[7] This paper builds on the notion of embodied cognition and human activity and applies them to the field of software or information modelling.

2. COGNITIVE APPROACHES
According to cognitive scientists George Lakoff and Mark Johnson some approached in the cognitive sciences have raised fundamental questions about the human faculty of "reason" that has been seen as independent of perception and bodily movement and separating us from other animals. Lakoff and Johnson suggest two ways in which evidence from the cognitive sciences differ from this view, firstly, they say, an evolutionary view suggests that human cognition, like animal cognition, grows out of bodily capacities and actions. Secondly, and perhaps more disquietingly, they are of the opinion that most cognitive functions such as hearing memory and learning have elements that are, for the most part, unconscious. Categorization, according to Lakoff and Johnson is an unconscious activity that characterizes every living being. All beings, they suggest, categorize food, predators, members of their own species and so on, based on their own sensorimotor abilities. The authors also go on to suggest that categorization, concepts and experience go hand in hand. For Lakoff and Johnson, categories and concepts are not pre-existing, objective reality - free and independent of mind and body but are

Conference'10, Month 1–2, 2010, City, State, Country.

formed throug bodily experience - and could therefore be different for different people. Thus human reality and reason, and human concepts in general are crucially shaped by our bodies and brains. Lakoff and Johnson offer the concepts of "Basic-level" and "Spatial" categorization as an example of embodied cognition. "Basic-level" categories according to them, are the highest level at which a single mental image can represent an entire category, for example, chairs, or cars. This is because it is difficult to get a single mental image of a generalized category such as "furniture". "Spatial" categories such as containment, over, under, at the back of, and so on, like basic-level categories, according to Lakoff and Johnson are unconscious and yet form a critical part of our perception of reality.[8]

2.1 Unconscious Influences in Design

According to Lakoff, categorization is implicit; automatic and unconscious and we are barely aware of it is, yet there is 'nothing more basic than categorization to our thought, perception, action, and speech'.[9] However, he suggests, categories do not evolve in a vacuum but develop side by side with widely held social views; this is what connects more abstract concepts to more real systems.[10]

Lakoff and Johnson also suggest that the conceptual system underlying our perspectives are often reflected in the metaphors we use. He gives the example of the concept of argument which is underlined by the metaphor of 'argument is war' - as reflected in statements such as 'the argument is *indefensible*' or '*attacks*' on weak arguments.[11] George Basalla, in his research on the evolution of technologies also highlights the use of metaphor. For Basalla, metaphors are crucial to any analysis, they are not, he suggests, 'ornaments arbitrarily superimposed on discourse for poetic purposes', but are relevant and 'at the heart of all extended analytical and critical thought'.[12]

Another factor that plays a role in cognitive functions is the probabilistic nature of sensory knowledge. The Bayesian theory of the Brain, for instance, suggests that the reliability of sensory knowledge is at best, a probabilistic one;[13] a familiar example of this is the snake-rope visual image, where the same object may appear as a snake or a rope, another example is driving in bad weather conditions. In both cases, the brain makes a decision based on probabilistic estimates. It therefore becomes important to specify, for example, in the case of the rope what it has been viewed as.

The use of specific metaphors, concepts and categories, implicit though they may be, points to an underlying framework of understanding that plays a role in cognitive activities. Contemporary empiricists suggest that it is only 'within the context of some background' that any observation or perspective may be considered valid. [14] When it comes to information architecture and design, there has been a tendency to view design as a somewhat mysterious process that is visible only once it has emerged from the mind of the designer. [15]

A notable effort to bring some of these factors to HCI (Human Computer Interface) design is by Manuel Imaz and David Benyon. Benyon and Imaz suggest the notion of domains and blends to help understand some of these concepts better. They describe a domain as a coherent sphere of activity observed and described in terms of the elements and relationships that make up the domain. A domain, according to them, might be very abstract, such as talking about the domain of love, traveling, or of games, or, it might be more concrete, such as talking about the domain of maternal love, sea journeys, or ball games. A metaphor, for Imaz and Benyon, is a cross-domain mapping, so that one domain is conceptualized in terms of the elements and relations of another. Metaphors, in their opinion, are central to our thought processes. Some of the metaphors encountered here are "The Human is an Information Processor" and "Argument is War". Software metaphors reflective of the change in methodologies from structured to object-oriented techniques are "Software is a sequence of steps" to "Software is a collection of objects". They distinguish an analogy from a metaphor; unlike an analogy which asserts that "something is like another thing", a metaphor, according to them, "establishes a true equivalence relationship"; one that suggests that 'A' *is* 'B'. Metaphors they suggest, are useful in that provide a way in which new interactions may be described in terms of previous experiences, or in other words, metaphors enable the description of new categories and relationships in terms of already known concepts and ideas.

Blends, are described by Imaz and Benyon as new ways of thinking and working that that emerge from a merging or *blending* of metaphors from distinct domains. The authors give the example of the 'desktop' metaphor from traditional office work-space, that was used to make the operation of the computer more intuitive. It was built on inputs from two different domains; the first, the domain of the office workspace with files, folders, desks, documents and even a trash can; the other, the set of computer commands such as print, copy, find, save and so on. Concepts from these domains *blended* to result in new ways of interaction characterized by icons, menus, pointing devices and other such that we now know as the 'computer' desktop. [16] Each metaphor is reflective of certain assumptions and constraints made in the domain.

In this situation, as information systems becomes increasingly complex, diverse, and globally distributed, there is also a need to uncover and make these influences more explicit. Especially since, despite decades of effort, the possibility of being able to identify standards of 'normal' cognitive function is, and may continue to be, an open question.[17] This paper attempts to seek mechanisms to include these influences in software models.

2.2 Cognitive Models

Imaz and Benyon, outline and apply the work of Lakoff and Johnson by describing Lakoff and Johnson's four distinct kinds of cognitive models.

a) Propositional models that specify the elements, their properties and the relations between them. This includes *frames, scenarios* and *scripts.* To give an example, the commerce *frame* may have the elements buyer, seller, goods and payment. A *script* describes a frequently occurring event-sequence. For example, a script for a birthday; scripts are usually used to avoid repetition. In the software world, a script usually relates to a sequence of steps that can be automatically executed by a person who 'runs' the script. A scenario, is described as a repetitive occurrence having an initial stage, a sequence of events and a final stage. An example could be encashing a cheque at a bank, where the starting step is of entering the bank and the final stage of leaving the bank with cash. Propositional models such as scenarios and scripts are used, often implicitly in software engineering generally in the form of usage scenarios (UML Use Cases) or test scripts. *Frames,* as a concept while not currently a part of the UML framework, though outside the scope of this discussion, may be seen as candidates for inclusion in UML.

b) Image schematic models that reflect representations based on bodily and cultural experiences. Examples of these include the Source-Path-Goal schema, the Containment schema and the link schema. Image schematic models

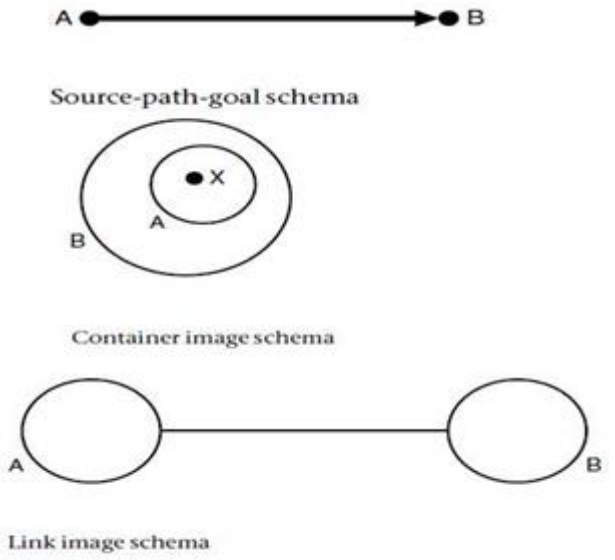

Figure 1: Image Schemata.

c) Metaphoric models: as discussed earlier, are derived from mappings from a model in one domain to another and form a part of the background of the system.

d) Metonymic models that are derived from the relationship between one or more element to the domain as a whole. Examples of this include statements like "Wall street may crash" where the place - wall street – is representative of the larger financial network.[18]

Thus far, the engagement in the paper has been with uncovering implicit and unconscious factors that influence technical modeling and design. In tune with a more embodied approach, the next section attempts to take the physical and social environment on board engages with the supposedly non-technical - social and physical environment - factors that nevertheless influence the conceptualization and implementation of technical systems.[19]

3. TOWARDS SOCIOTECHNICAL MODELLING

It has been suggested that the increasing dependence of human beings on technology has led to a tendency to seek technical solutions to social or human problems.[20] This translation from the social to the technical may be seen as underlying the requirements gathering and analysis and design phases of the software lifecycle. The approach in this paper is to be more inclusive rather than keep the social and technical realms distinct from each other. This is more in tune with the views of Michel Callon and his co-authors who have suggested that 'the distinction between what was 'social' and 'technical' and what was 'human' and 'non-human' was itself disrupted through the process of technical change'. It was not therefore possible to give a 'purely "social" explanation of technical change because technical objects (facts, artefacts, devices) themselves formed a critical part of what the social is.[21] The software development process is seen as sociotechnical one, involving discussions, debates and engagements amongst multiple voices. These may include the voices of various user groups and stake holders apart from differing opinions and perspectives on design and technology choices.

It has been suggested that each technological systems masks the historicity of interests and the politics of the specific technology. Sociologist of technology, Andrew Feenberg, argues that each technology embodies some particular people's voices and is a reflection of certain relations of power. More often than not, these factors influence which among many possible considerations are to be included and which are to be excluded in the shaping of technology. A larger social involvement, according to Feenberg, entails inclusion of a larger section of users and stake holders and an active engagement with questions of who determines choices, at what cost, to what end and by what institutional or social process. [22]

It would also be relevant to note the distinction between voices and agency. Here, voices refers to the capability to make one's views heard. Which voices are considered germane to the task at hand needs to decided by some well defined, known criteria. Agency refers to the capability to influence decision-making, agencies could be individuals, institutions, and human or natural artefacts and events. [23]

3.1 Values in Information Systems

Information or software systems share two characteristics not commonly found in other engineering fields. Firstly, their end products or artefacts are not physical objects but data, symbols and images that help in cognitive processes. Secondly, the text-book phases of the Software Development Life Cycle (SDLC) tend to be overlapping and iterative. These characteristics together suggest that values in software systems need a somewhat different engagement. In a study conducted to see if software professionals see the stages of development, maintenance and support of software as distinct or overlapping, most people most people were of the view that the tree are a part of the same continuum.[24] This general trend seems to be borne out by available literature.[25]

This overlap becomes especially relevant when considering the role of values in software systems since the values prevalent during the conceptualization, development and maintenance phases tends to extend to the system being developed. In other words, it may not be feasible to produce a system that is meant to provide transparency, by being opaque in the process of development.

Thus it becomes relevant to identify the values that the system intends to uphold and promote. Once identified, there is a need to identify criteria that help to measure it and mechanisms to acquire the metrics.

3.2 Project Overview Diagram

Summarizing the discussion till now, traditional (disembodied) cognitive models, by approaching the brain as an abstract goal-oriented system have been related to software systems that also use symbols to create abstract models of real-world systems. In the object-oriented paradigm, these are represented through figures such as UML class diagrams, sequence diagrams and so on. More embodied approaches such as those outlined by Lakoff and Johnson and extended by Benyon and Imaz to HCI, also suggest different conceptualizations and representations of the UML. At the root of each such exercise is software Project. Each project involves one or more domains of knowledge which are associated with some key concepts, assumptions and constraints.

The model also includes the implicit, often unconscious categories, concepts, metaphors, blends and metonyms that may be associated with one or more domains. In addition, the blurring of the boundary between the social and the technical also meant that supposedly, non-technical factors also effect information architecture and design. It therefore becomes important to identify the voices and agencies that include constitute individual, institutions, technical, social and natural aspects. Values are intrinsic to the project and tend to get reflected in the deployed system. They are important to identify as are the criteria and mechanisms to measure them.

This description is represented in the figure below. The stereotypes in the figure (indicated by the guillemots "<<" and ">>") represent the additions that form a part of the UML extension.

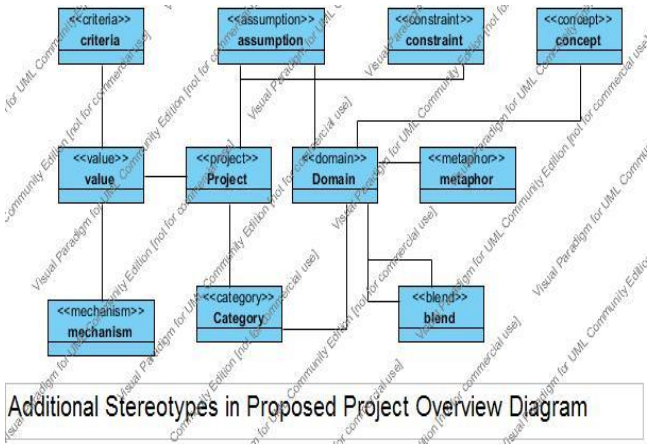

Figure 2: Additional Stereotypes in Project Overview Diagram

Note: The basic categories identified with a domain are akin to UML classes.

The role of information systems as aids or enablers of cognitive activities was briefly mentioned earlier. The next section uses the field of 'activity theory' to understand how the diverse activities from different domain areas may be abstracted and modelled.

4. MODELLING 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 seen as mediators between 'internal' mental processes and the external world.[26] Indeed, activity theory suggests that that the tools used to perform an activity gradually become a part of 'internalized' 'mental' processes.[27] 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.[28] 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:

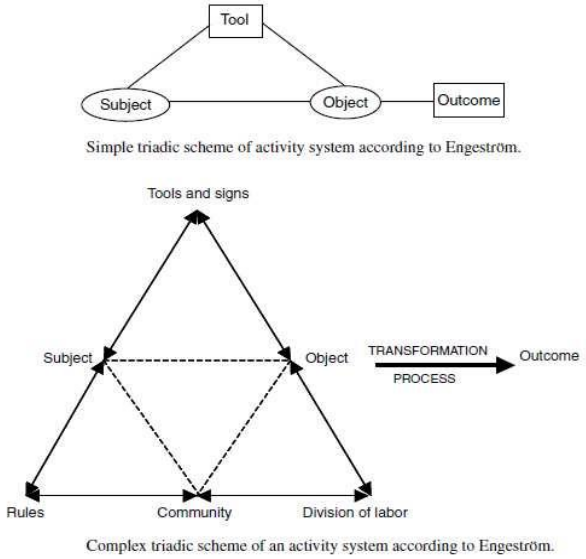

Figure 3: Activity Schemas [29]

4.1 Activity-Context Diagram

The activity-context diagram combines the *context diagram from* structured analysis and design - that gives 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 - and the *Activity Schema* as shown above. The next section shows a sample of an activity-context diagram.

5. SAMPLE USAGE

A sample use of the two proposed diagrams for the Hospital Billing 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 treatment, with the associated billing and support actions tasks. These are performed by Doctors, Nurses, and other Hospital Staff who use a variety of technical tools to help them in their tasks. This brief overview is represented in the following activity context diagram. It should be noted that this is only a sample to provide an overview, it does not claim to represent a what the actual elements of a hospital billing system would include.

5.1 The Hospital Billing Activity-Context Diagram

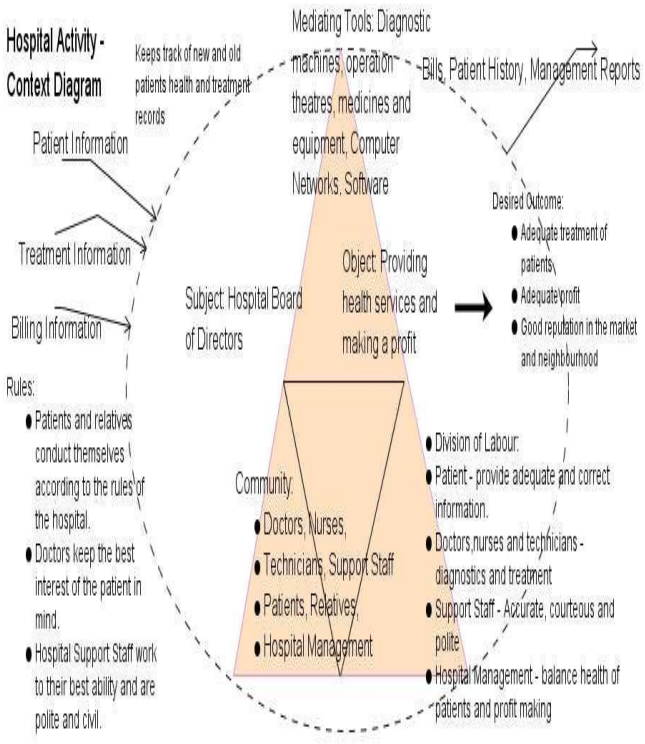

Figure 4: Hospital - Ac**tivity Context Diagram**

5.1.1 Elements of the Hospital Billing Activity-Context Diagram

- One line description: the system keeps track of patient health and treatment records.
- Inputs to the system: Patient information, Treatment information, Billing information
- Outputs of the system: Bills, Patient history, Management Reports
- Subject or Activity Owner: Hospital Board of Directors
- Object: Providing health services and making a profit
- Expected or Desired Outcome: Adequate treatment of patients, Adequate Profit, Good reputation in the Market and Neighbourhood
- Tools: Diagnostic tools, Operation theatres, Medicines, Computer networks and Software
- Community: Doctors, Nurses, Patients and Relatives, Support staff and the Hospital Management
- Rules: Patients and relatives conduct themselves according to the rules of the hospital, Doctors keep the best interest of the patient at heart, Support staff work sincerely and are civil and polite
- Division of Labour: Patients provide complete and correct information, Doctors are responsible for diagnosis and treatment, Support staff to provide courteous and timely services and Hospital Management to draw a balance between providing efficient services and profit making.

5.2 The Hospital Billing Project Overview Diagram

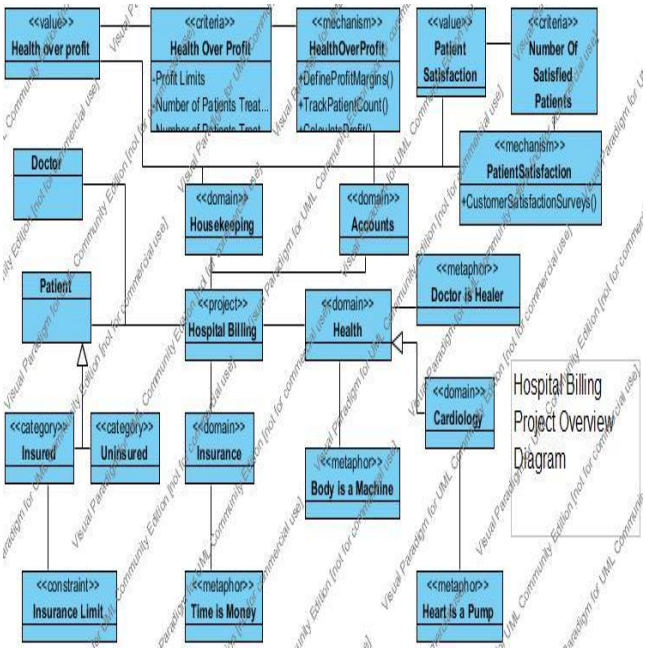

Figure 5: Hospital Billing Project Overview Diagram

5.2.1 Elements of the Hospital Billing Project Overview Diagram

- The project draws upon the domains of Health, Insurance, Accounts and House Keeping. Since a domain may include other domains, the domain of cardiology is included under health as an example.
- Doctor and Patient are UML Classes.
- The Categories or insured and uninsured inherit from the Patient Class
- The metaphor "Doctor is Healer" is associated with the domain of Health, the metaphor of the heart as a pump is associated with the Cardiology domain and the "time is money" metaphor describes the insurance domain.
- The values associated with the Hospital Billing Project are those of "Health Over Profit" and "Patient Satisfaction", the criteria associated the first involve defining profit margins and the number of insured and uninsured patients treated, patient satisfaction is measured by the number of satisfied patients. Both sets of criteria are associated with the accounts and house keeping domains.

6. CONCLUSION

Software development in recent years is becoming increasingly domain and context specific. At the same time there is a need for a shared across globally distributed collaborations and activities. These aspects may be addressed by finding mechanisms to identify and include these factors in the software model. While this paper does not claim to provide an exhaustive mechanism to do so, however, it does claim to provide a starting point. The activity-context and the problem-overview diagrams described in the paper are tools that I have found useful starting points in design. In accordance with UML tradition, I am submitting them as possibilities for wider use.

7. REFERENCES

[1] Rieke Fred, Warland David, Van Steveninck Ruyter Rob de, and Bialek William, 1999. *Spikes:Exploring the Neural Code*. The MIT Press. Cambridge, Mass. pp 1-5.

[2] Neumann John Von, 1958. *The Computer and the Brain*, Yale University Press, New Haven, London, page 43.

[3] Anderson John, 2007. *How can the human mind exist in a physical universe?* Oxford University Press, Oxford, New York, pp 7-8.

[4] Varela Francisco, Thompson Evan, Rosch Eleanor, 1991. *The Embodied Mind: Cognitive Science and Human Experience*, The MIT Press, Massachusetts, pp 150-180.

[5] Hayles Katherine N, 1999. *How we became post human: Virtual Bodies in Cybernetics, Literature, and Informatics*, The University of Chicago Press, Chicago, London. pp 1-10.

[6] Gonzalez Quilici Maria Eunice, 2008. 'Information and mechanical models of intelligence: What we can learn from cognitive science', in Dror Itiel E, Harnad Steve Ed, *Cognitive Distribution: How cognitive technology extends our minds*, John Benjamins Publishing Company, Amsterdam/Philadelphia, pp 109-114.

[7] Varela Francisco, Thompson Evan, Rosch Eleanor, 1991. *The Embodied Mind: Cognitive Science and Human Experience*, The MIT Press, Massachusetts, pp 150-180.

[8] Lakoff George, Johnson Mark. 1999. *Philosophy in the flesh: The Embodied Mind And Its Challenge To Western Thought*, Basic Books, New York, NY. pp 26-50.

[9] Lakoff George, 1987. *Women, Fire and Dangerous things: what categories reveal about the human mind*, University of Chicago Press, Chicago, page 6.

[10] *Ibid*, pp 157-158.

[11] Lakoff George, Johson Mark. 2003. Third Edition (First Edition 1980). *Metaphors We Live By*, University of Chicago Press. London. pp 4-8.

[12] George Basalla, 1998. *The Evolution of technology*, Cambridge University Press, Cambridge, UK, page 3.

[13] Knill David C, Pouget Alexandre, 2004. *The Bayesian brain: the role of uncertainty in neural coding and computation*, Trends in Neurosciences, Vol 27 No 12, Dec 2004, pp 712-714.

[14] Tanesini Alessandra, 1999. *An introduction to Feminist Epistemologies*, Blackwell Publishers, Malden Mass, Oxford UK, pp 28-29.

[15] Kolko Jon, 2011. *Exposing the Magic of Design: A Practitioner's Guide to the Methods and Theory of Synthesis*, Oxford University Press, Oxford, UK, New York, USA, page xi.

[16] Imaz Manuel, Benyon David, 2007. *Designing with Blends: Conceptual Foundations of Human-Computer Interaction and Software Engineering*, The MIT Press, Cambridge, Mass, London, England, pp 36-60.

[17] Frances Allen, 2009 . 'Whither DSM-V', *The British Journal of Psychiatry*, October. 195, 2009, pp 391-392.

[18] Imaz Manuel, Benyon David, 2007. *Designing with Blends: Conceptual Foundations of Human-Computer Interaction and Software Engineering*, The MIT Press, Cambridge, Mass, London, England, pp 1-35.

[19] Latour Bruno, *Reassembling the Social: An introduction to Actor-Network Theory*, Oxford University Press, New York, 2005, page 1-7.

[20] Winner, Langdon: *Autonomous Technology*, Cambridge, Massachusetts and London, England: The MIT Press,1977. pp 10-17.

[21] Barry Andrew, Slater Don, 2002. 'Introduction: The technological economy', *Economy and Society*, Routledge, London, Volume 31 Number 2, May 2002. Page 177.

[22] Gerald Doppelt, 2006. 'Democracy and Technology', in Tyler Veak Ed, *Democratizing Technology: Andrew Feenberg's Critical Theory of Technology*, State University of Albany Press, Albany, New York. pp 85-87.

[23] Kaptelinin Victor, Nardi Bonnie A, *Acting with Technology: Activity Theory and Interaction Design*, The MIT Press, Cambridge, Mass, London, England. 2006. Page 238

[24] See Software Development and maintenance processes at http://stasis.in/Software-Dev-Maint-Support Accessed: 14 Mar 2014.

[25] Basili Victor R, *Viewing Maintenance as Reuse-Oriented Software Development*, IEEE Software, January 1990, pp 19-25.

[26] 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.

[27] Kaptelinin Victor, Nardi Bonnie A, *Acting with Technology: Activity Theory and Interaction Design*, The MIT Press, Cambridge, Mass, London, England. 2006. pp 42-44.

[28] *Ibid*, page 196.

[29] 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.