Towards Socially Embedded Software Engineering

—— Introducing Social UML ——

Ravi Shukla (Jawaharlal Nehru University, New Delhi)

Abstract The typical software engineering process entails translating real-world social or human problems to technical ones. The process of translation and the subsequent analysis, design and implementation steps tend to exclude social 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. Background

With the increasing use of software systems in society, there is a need to include more social aspects in the software engineering process. This need is seen in approaches such as Joint application and design (JAD) sessions where end-users and the IT team work together to define the business requirements (Haag, Cummings: 2004)[i], and in requirement modeling frameworks such as i* (Yu: 2009)[ii]. Other related approaches include 'agent oriented approaches' such as the Agent Modeling Language (AML) that attempts to extend the Unified Modeling Language (UML) (Cervenka et al: 2009)[iii]. These models seem to focus either on requirement specifications or on the interactions between software agents and may not suffice in including sociological concepts in software development. Building on the concepts of Cultural Historical Activity Theory (CHAT) and Interaction Design, I build a case for Social UML (SUML), an extension of UML that includes social concepts and ideas in the process of software design and documentation. Even a brief overview of either field is outside the scope of this paper, however reference for the first (Blunden:2009, Bedny et al: 2007, Kaptelinin:2006)[iv] and second (Yatco:2012, Kaptelinin:2006, Imaz etc al, 2007, Kolko:2011)[v] subjects may be found at the end of the paper.

2. Introducing Social UML

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.[vi] 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. A case study to provide some sample diagrams follows.

2.1 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.

2.2 Diagrams Proposed

2.2.1 Activity-Context Diagram

This in conceptually a blend of the *context diagram* in traditional structured analysis and design and the Activity Triangle model of *Activity theory* as described by Engestrom. The blended diagram is shown for the case study at the end (Fig 2).

2.2.2 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 diagram is shown for the case study at the end (Fig 3).

2.3 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.

2.3.1 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.

2.3.2 Types of Time

Depending on the nature the system under consideration, different notions of temporality may be relevant. As a space saving device, the diagrammatic representation of different notions of time is shown below in Fig 1 at the end

2.4 Additional Elements and Attributes (profiles)

In order to include notions of temporality, the following elements and attributes may be added to existing elements in UML. Their usage is described in 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.

- 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.

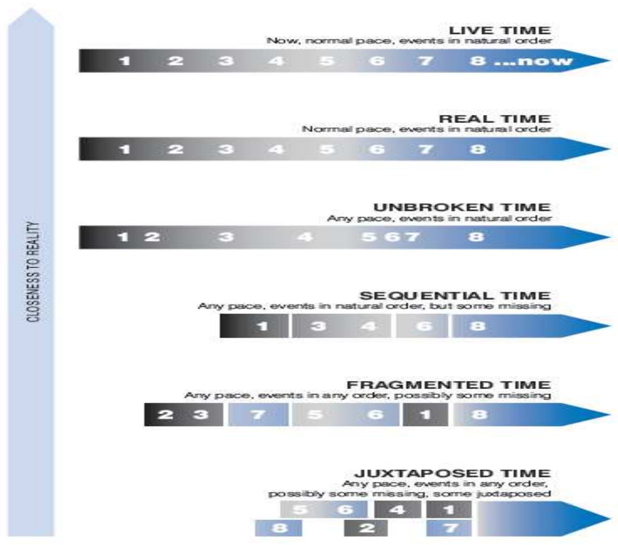
- Archival Procedure – What is the procedure to be followed when the expected life span ends.
- Exit Procedure – What the is the procude to be followed for changing the status of the object.

3) Case Study

The case study of a Hospital Information System in 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

4) Figures

Fig 1: Different notions of time

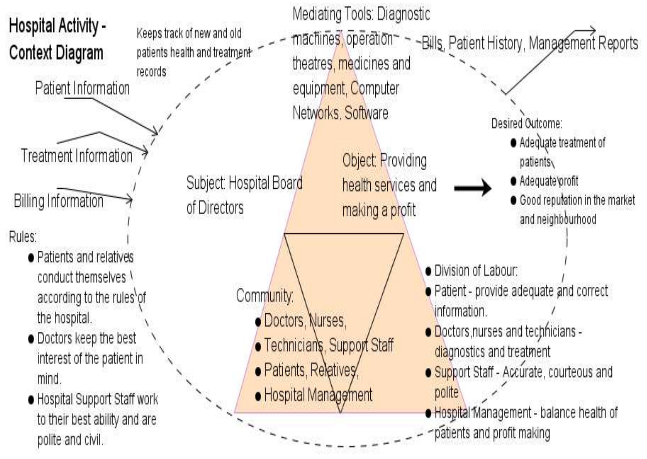


CLOSENESS TO REALITY
LIVE TIME
Now, normal pace, events in natural order
1 2 3 4 5 6 7 8 ...now
REAL TIME
Normal pace, events in natural order
1 2 3 4 5 6 7 8
UNBROKEN TIME
Any pace, events in natural order
1 2 3 4 5 6 7 8
SEQUENTIAL TIME
Any pace, events in natural order, but some missing
1 3 4 6 8
FRAGMENTED TIME
Any pace, events in any order, possibly some missing
2 3 7 5 6 1 8
JUXTAPOSED TIME
Any pace, events in any order, possibly some missing, some juxtaposed
5 6 4 1
8 2 7

Lundgren Sus, Hultberg Theo, ‘Time, Temporality and Interaction’, *Interactions,* July + August 2009, Association for Computing Machinery (ACM), pp 34-37.

The case study diagrams follow.

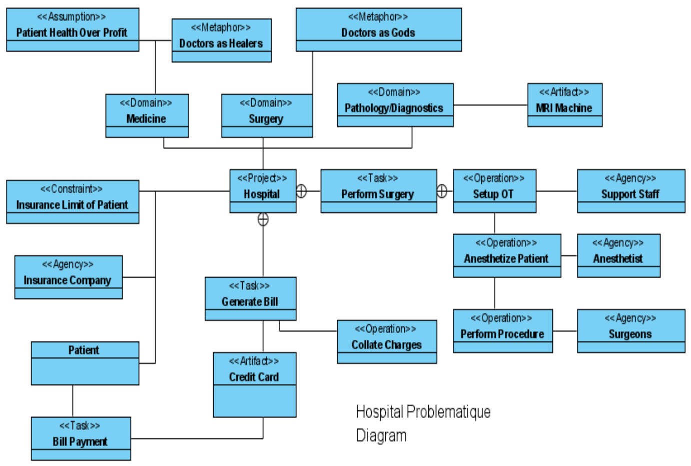
Fig 2: Hospital Activity-Context Diagram



Hospital Activity - Context Diagram
Keeps track of new and old patients health and treatment records
Patient Information
Treatment Information
Billing Information
Rules:
Patients and relatives conduct themselves according to the rules of the hospital.
Doctors keep the best interest of the patient in mind.
Hospital Support Staff work to their best ability and are polite and civil.
Mediating Tools: Diagnostic machines, operation theatres, medicines and equipment, Computer Networks, Software
Bills, Patient History, Management Reports
Subject: Hospital Board of Directors
Object: Providing health services and making a profit
Desired Outcome:
Adequate treatment of patients
Adequate profit
Good reputation in the market and neighbourhood
Community:
Doctors, Nurses,
Technicians, Support Staff
Patients, Relatives,
Hospital Management
Division of Labour:
Patient - provide adequate and correct information.
Doctors,nurses and technicians - diagnostics and treatment
Support Staff - Accurate, courteous and polite
Hospital Management - balance health of patients and profit making

The diagram in Fig 2 represents tasks. Each that tasks can comprise a number of sub-tasks or operations. These are not complete representations but are only presented here to present the idea.

Fig 3: Hospital Problematique Diagram



<<Assumption>> Patient Health Over Profit
<<Metaphor>> Doctors as Healers
<<Metaphor>> Doctors as Gods
<<Domain>> Medicine
<<Domain>> Surgery
<<Domain>> Pathology/Diagnostics
<<Artifact>> MRI Machine
<<Constraint>> Insurance Limit of Patient
<<Project>> Hospital
<<Task>> Perform Surgery
<<Operation>> Setup OT
<<Agency>> Support Staff
<<Operation>> Anesthetize Patient
<<Agency>> Anesthetist
<<Agency>> Insurance Company
<<Task>> Generate Bill
<<Operation>> Collate Charges
<<Operation>> Perform Procedure
<<Agency>> Surgeons
Patient
<<Artifact>> Credit Card
<<Task>> Bill Payment
Hospital Problematique Diagram

5) Conclusion

The inclusion of socialogical aspects in software engineerig extends beyond user participation; by including these aspects, it may be possible to build software more in tune with social values and asiprations.Extending UML to take note of these concepts enables the inclusion of social voices and agencies into the processe of requirements, analysis and design without making them mandatory so that they may be used as required.

References

[i] Haag Stephen, Cummings Maeve, *Management Information Systems for the Information Age’,* Mc Graw Hill Publications, New York, 2004, pp 289-293.

[ii] Yu, Eric S, ‘Social Modeling and i*’, Borgida, A.T. et al. (Eds.), *Conceptual Modeling: Foundations and Applications, Essays in Honor of John Mylopoulos,* Lecture Notes in Computer Science, Vol 5600, Springer-Verlag, Heidelberg, Berlin, 2009. pp 99-100.

[iii] 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, Heidelberg, Berlin 2009. pp 28-30.

[iv] Blunden Andy, *The Interdisciplinary Theory of Activity*, Brill Publishing, Leiden, Netherlands, 2010.

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.

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

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

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

[vi] 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, London, Moscow, 2003. pp 8-10.