Beg, Borrow or Steal: OK, but it's not all One-Way Traffic!

G. Michael McGrath
JRCASE, Division of ICS
Macquarie University
Sydney, NSW, 2109
Australia
+61 2 9850 9581
mmcgrath@ics.mq.edu.au

Lorna Uden
School of Computing
Staffordshire University
The Octagon, Beconside
Stafford, ST18-OAD, England, UK
+44 1785 353276
L.Uden@staffs.ac.uk

Abstract: *Just as the social science literature is rich in models relevant to the software development process, these fields could benefit greatly from advances in knowledge representation methods used within software engineering and related disciplines. Here, we advocate the use of formal conceptual modelling techniques, commonly employed in software engineering work, for representing models and theories developed within the social sciences and, especially, management science fields. We illustrate our thesis through a brief description of an instance where a formal representation of activity theory was employed to capture critical 'softer' factors within a system development project.*

Keywords and Phrases: Formal conceptual models, softer issues, activity theory.

1. Domain: Capturing 'Softer' Factors in Software Engineering Research

The evidence suggests that the IS implementation success rate is poor. According to Rodriques and Bowers [12], budget over-runs of 40-200% are becoming increasingly common; a Standish Group study report claims that 30% of projects are cancelled prior to completion, over 50% of completed projects cost almost twice their estimated cost and, in 1995, US organizations spent $81 billion on cancelled projects and an additional $59 billion on project over-runs [13]; and, finally, a report prepared by U.K. Organisational Aspects Special Interest Group (OASIG) [11], indicates that around 80% of new systems are delivered late and over-budget, and 80-90% of IS investments do not meet their performance objectives. The OASIG report goes on to state that people-related, 'softer' problems are frequently the major cause of these failures. Softer problems encompass aspects such as leadership, motivation, morale, organization culture, and power and politics [2].

In this context, *activity theory* would appear to have much to offer. Activity theory incorporates notions of intentionality, history, mediation, motivation, understanding, culture and community. These aspects of activity theory have proved attractive to human-computer interaction (HCI) researchers as an alternative framework to the dominant cognitive psychology paradigm [10]. In particular, activity theory provides a framework in which the critical issue of *context* can be taken into account. Thus, in extending their horizons, the HCI community has moved into the realm of process modelling, systems analysis and, particularly, requirements elicitation. Unfortunately, however, a perusal of the activity theory literature reveals very little in the way of prescriptive guidelines for its application. Our problem then is how can we best take advantage of the many attractive features inherent in activity theory identified above and do so in a way that might be beneficial to both researchers and practitioners?

This issue came to a head when we were invited to undertake a retrospective analysis of a system implementation that had been less than successful. The organization concerned was a large Australian company in the information services business and we were asked to focus on their efforts to build an 'intelligent help system' (IHS), in order to alleviate some significant problems they were experiencing with their product provisioning and billing processes. The research team was promised full access to all relevant project documentation and personnel.

As far as we can tell, this commitment was honoured. The work was undertaken as an exploratory case study and it was decided to collect and analyse data using an activity theory framework. We now briefly introduce activity theory and our framework and then discuss benefits and limitations with our approach identified during the study[1]. One conclusion we did reach was that, if the software engineering community is to take full advantage of activity theory, a more formal exposition is required. We generalise from this and conclude our paper by arguing that, while software developers would do well to take greater note of the rich variety of relevant material within the social sciences and organization and management theory (OMT) literature, researchers in these fields might do equally well to utilise conceptual modelling techniques commonly employed in software engineering applications and research.

2. Activity Theory and SATBPA

Activity theory was originated by the Russian psychologist Lev Vygotsky [14] (and colleagues) around 1930. As noted previously, more recently, the theory has been used in the area of HCI design [9]. Here, we briefly introduce activity theory and our own interpretation of some key elements of the theory. For a detailed introduction, the reader is referred to [7].

An *activity* is seen as a system of human 'doing' whereby a *subject* works on an *object* in order to obtain a desired *outcome* (see Figure 1). In order to do this, the subject employs *tools.* A subject may be an individual or a composite entity (e.g. a company, an organization unit or a labour union). Many subjects may be involved in an activity, and each subject may be involved in one or more roles and have multiple *motives. Rules* (which include norms, values, beliefs etc.) are both explicit and implicit, and specify how subjects fit into communities and how work is broken up between subjects. An activity may be broken down into *actions* and actions, in turn, may be decomposed into *operations.* Actions are well-defined processes directed at specific, conscious *goals,* while operations may be thought of as (largely) automatic responses to prevailing *conditions.* Activities, in time, may become actions and actions, when well-entrenched, may become operations. Movement in the other direction within this hierarchy is also possible. Finally, one subject's operation may be another's action or activity (where, for example, the former is experienced and the latter is a new hire).

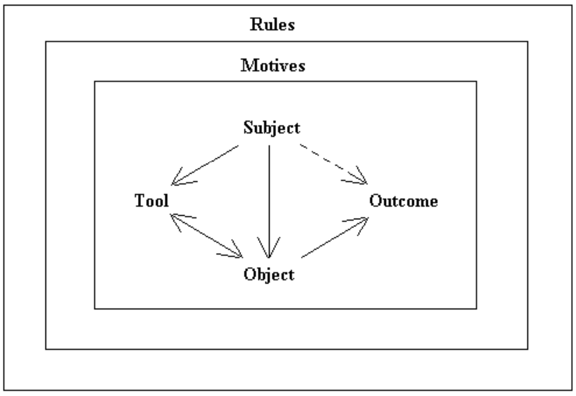

Figure 1: Activity components

A fairly generic example of a business activity (taken from our case study) is 'satisfy customer need'. Subjects involved in the activity include service staff and senior management. One of the significant objects of the activity is a product order, and tools used to assist subjects include documented procedures, an order-entry information system and performance reports. Outcomes are closely related to motives and these vary with subjects. For example, our research revealed that some staff were motivated by both a desire to deal with customers as quickly as possible and a fear of losing their jobs. Some senior management, on the other hand, were motivated very much by a perceived need to improve the efficiency of the activity. Rules include well-documented specifications of which subject does what and when, as well as beliefs such as "senior management want to take our jobs away" (a common view among service staff in our study).

[1] Space limitations do not permit a detailed description of the case study itself but an (as yet unpublished) account will be made available to Workshop participants.

In our view, rules and their ability to represent motives, norms, beliefs, values etc.) are quite possibly activity theory's most useful feature. For example, using these, researchers can identify contradictory beliefs and values, motive and goal conflicts between different individuals and communities, and contention clashes for resources. The difficulty is that, in any non-trivial study, maximum benefit from this analysis will only result if some sort of highly-structured (and, preferably, automated) data storage and analysis approach is employed.

To this end, we have developed *SATBPA (System for Activity Theory Business Process Analysis).* SATBPA is an automated tool that provides an interface to a database designed to record the detail of business processes in a form consistent with activity theory. This work has been described in more detail in [9], where we remarked that SATBPA is intended to be a first step towards a the development of a methodology for the structured application of activity theory in modelling business processes (for research, actual information systems development and other purposes). The SATBPA repository is specified conceptually at both the data and meta-data levels in entity-relationship form. The repository itself has been implemented in Microsoft *Access*™, interfaced with a Prolog-based expert system, and our rules, discussed earlier, are constructed in terms of the contents of domain-specific (e.g. case study) repositories.

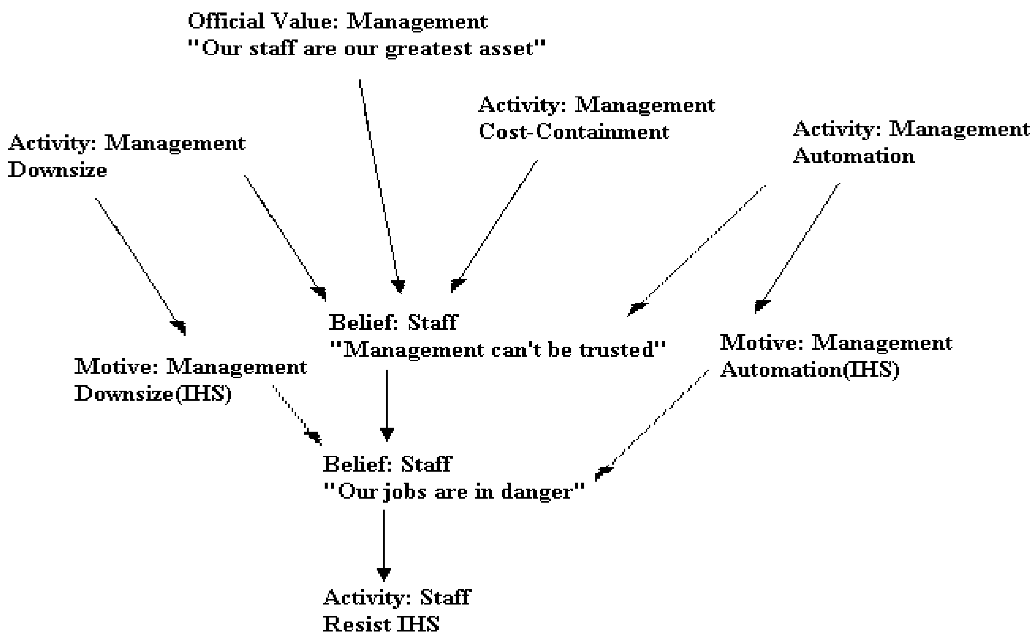

Figure 2: Derivation of a case study conclusion

An illustration of how our SATBPA framework was employed to derive a potential source of resistance to the implementation of IHS is presented in Figure 2. The organization's management frequently declared "Our staff are our greatest asset": effectively, an official company value. At the same time, they were vigorously pursuing downsizing, automation and other cost-containment initiatives. As a result, service staff were extremely cynical of this often-espoused company value and, instead, arrived at their own (unofficial) belief that "Management are out to nail us and can't be trusted". Thus, when IHS was proposed, this belief, along with the staff's (assumed) knowledge of management's downsizing and automation motives, led them to the belief that their jobs were in danger. In turn, this resulted in resistance to the development, implementation and use of the IHS system.

The above reasoning chain was derived: from facts and rules explicitly specified in our SATBPA repository; from rules induced from repository contents; and, ultimately, by deductive inference. Space limitations prevent a comprehensive treatment of our conceptual modelling scheme and inference approach, details of which can be found in [9]. For the purposes of this paper, the essential aspect is that the use of a *formal* conceptual modelling scheme facilitates collection, storage and automated analysis of case study data.

3. Activity Theory and Software Engineering Research: Strengths and Weaknesses

A strength of activity theory, when employed in software engineering case study research, is that it virtually demands that the investigator does not neglect the critical softer factors. In the previous section, we presented an example of user resistance to an information systems implementation, motivated by a distrust of management and suspicion of their perceived objectives. Many other important issues that might be classified in the 'softer' category were revealed during our case study. These included: i) internal performance appraisal measures inconsistent with good customer service and relations; ii) considerable staff frustration at a perceived lack of involvement in requirements elicitation; and iii) major discrepancies between management and staff norms and values (and between stated and actual values).

As noted, however, the major difficulty we encountered with activity theory, is an almost complete lack of any prescriptive guidelines on how it should be applied in the field. Recently, we have noted the emergence of an increasing number of papers devoted to activity theory. Many case studies have been reported (see e.g. [5, 10]) and a wide variety of techniques have been employed for data collection, organization and analysis. Most of these case descriptions are narrative in style and, while many are extremely interesting, we have experienced considerable difficulty in our attempts to map the essential features of these studies back to the wonderfully elegant exposition of an activity originally presented by Engestrom [4] and, later, expanded by Kuutti [7].

We are unabashed proponents of a much more structured application of activity theory than is currently found in the literature and, moreover, we argue in favour of the Engestrom/Kuutti representation as the appropriate starting point. This is the case with our own approach, where we take Kuutti's structure and map it to entity-relationship form, supplemented with rules expressed in first-order logic. Essentially, we are advocating the use of formal conceptual modelling techniques commonly employed in software engineering work for representing models and theories developed within the social sciences and, especially, management science fields. That is, just as the OMT and social science literature is rich in models relevant to the software development process, these fields could benefit greatly from advances in knowledge representation methods used within software engineering and related disciplines.

We are reminded here of Jaques [6] who has argued that, despite massive efforts by organization and social science experts, only the merest beginnings of an organization and management science have become evident. In arguing his case, he reserves particular criticism for *"the pile of vague and ill-defined terms that litter the field"* and notes that *"Without --- clear meaning it is impossible to think, or to test propositions, or to talk to one another with any hope of understanding"* (p. 10). These observations are consistent with an increasing body of OMT research that has addressed the benefits of formal models, as opposed to informal, literary theorising. For example, Bendor and Moe [1] claim that more formal models clarify chains of reasoning and simplify the task of verifying that conclusions do indeed follow from assumptions; McGrath [8] points to the clarification of concept overlaps, ambiguities and inconsistencies; and Curtis, Kellner and Over [3] suggest that the properties of multiple model perspectives can be analysed more accurately where representation constructs are formally constrained.

Finally, to return to our starting point, the ever-growing range of computing and communications technologies generally plays only an enabling role in the development and utilisation of the systems designed to achieve an end-user's business objectives. The building of information systems involves, at its core, the still emerging discipline of software engineering as well as other more traditional areas of engineering and science and a number of non-technical management, socia disciplines. In essence, systems are not built in a vacuum, but within organizational environments where outcomes are heavily influenced by a myriad variety of internal and external softer factors. Modelling both 'soft' and 'hard' data using a common approach reduces the possibility that the critical softer factors will be overlooked.

References

1. Bendor, J. and Moe, T.M. Bureaucracy and subgovernments - a simulation model. (in Masuch, M. and Warglien, M. eds.), *Artificial Intelligence in Organization and Management Theory.* North-Holland, Amsterdam, 1992, 119-141.

2. Constantine, L.L. *Constantine on Peopleware.* Prentice-Hall, Englewood Cliffs, NJ, 1995.

3. Curtis, B., Kellner, M.I. and Over, J. Process modelling. *Communications of the ACM,* (35, 9), 1992, 75-90.

4. Engestrom, Y. *Learning by Expanding.* Orienta-Konsultit, Helsinki, 1987.

5. Engestrom, Y., Miettinen, R. and Punamaki, R.L. *Perspectives on Activity Theory.* Cambridge University Press, UK, 1999.

6. Jaques, E. *Requisite Organization.* Cason-Hall, Arlington, 1996.

7. Kuutti, K. Activity theory as a potential framework for Human-Computer Interaction research. (in Nardi, Bonnie A. ed.), *Context and Consciousness: Activity Theory and Human-Computer Interaction*. MIT Press, Cambridge, MA, 1996, 17-44.

8. McGrath, G.M. MP/L1: Towards an automated model of organisational power. *Proceedings of the 2nd IEEE Australian and New Zealand Conference on Intelligent Information Systems,* (Brisbane, 29 Nov. - 2 Dec., 1994), 487-491.

9. McGrath, G.M. and Uden, L. Modelling "softer" aspects of the software development process: An activity theory based approach. *Proceedings of the 33rd Hawaii International Conference on System Sciences (Software Process Improvement Minitrack),* (Maui, Hawaii, 4-7 Jan., 2000).

10. Nardi, Bonnie A. *Context and Consciousness: Activity Theory and Human-Computer Interaction.* MIT Press, Cambridge, MA, 1996.

11. OASIG. The performance of information technology and the role of human and organizational factors. *Report to the Economic and Social Research Council, UK,* available through URL: http://www.shef.ac.uk/~iwp/resprogs/iperf.html, 1996.

12. Rodriques, A.G. and Bowers, J. System dynamics in project management: A comparative analysis with traditional methods. *System Dynamics Review,* (12, 2), 1996, 121-138.

13. Standish. Chaos. Available through: http://www.standishgroup.com/chaos.html, 1995.

14. Vygotsky, L. S. *Mind and Society*. Harvard University Press, Cambridge MA, 1978.

BRIEF BIOGRAPHIES

Dr. G. Michael McGrath gained his PhD from Macquarie University, Australia, in 1993. He is currently Deputy Director of the CSIRO-Macquarie University Joint Research Centre for Advanced Systems Engineering (JRCASE), where he heads a research strand focusing on socio-technical aspects of systems and software engineering. He has over 20 years experience in the IT industry - mostly at Telstra, Australia, where he worked in a variety of positions. These included Senior Project Manager, responsible for the development of Telstra's multi-million dollar supply systems applications, and an executive-level position, as Manager Information Architecture within the organisation's Corporate Strategy Directorate. Much of his recent research has been directed towards the development of a process modelling framework that incorporates the critical 'softer' or people-related issues notably absent from most process models. Models, mostly derived from the organisation and management theory literature, are represented using formal conceptual modelling techniques commonly employed in information systems work. This facilitates the comparison of different theories in the same domain – particularly, the identification of concept overlaps, ambiguities and inconsistencies. Areas tackled to date include organisational power and politics, motivation and leadership.

Lorna Uden is a Senior Lecturer in computing science at the School of Computing, Staffordshire University, UK, where she has recently completed her PhD program. Her research interests include Technology in Learning, Constructivism, Problem-based Learning, Software Engineering, Human-Computer Interaction, Object-Oriented Design, Multimedia, Internet Learning, CSCW and Activity Theory. She has published over 20 journal and conference papers on these topics and has developed a Courseware Methodology for technology-based learning.