

University of Wollongong
Research Online

Faculty of Engineering and Information
Sciences - Papers: Part A

Faculty of Engineering and Information
Sciences

2005

Eliciting and specifying requirements for highly interactive systems using activity theory

Robert B. K. Brown
University of Wollongong, bobbrown@uow.edu.au

Peter Hyland
University of Wollongong, phyland@uow.edu.au

Ian C. Piper
University of Wollongong, ian@uow.edu.au

Follow this and additional works at: <https://ro.uow.edu.au/eispapers>



Part of the [Engineering Commons](#), and the [Science and Technology Studies Commons](#)

Recommended Citation

Brown, Robert B. K.; Hyland, Peter; and Piper, Ian C., "Eliciting and specifying requirements for highly interactive systems using activity theory" (2005). *Faculty of Engineering and Information Sciences - Papers: Part A*. 930.
<https://ro.uow.edu.au/eispapers/930>

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au

Eliciting and specifying requirements for highly interactive systems using activity theory

Abstract

The processes of eliciting user requirements and formalising these into specifications are critical for the success of highly interactive systems. These processes are still poorly understood, partly because current methods are usually ad hoc and lack any theoretical basis. A number of researchers have used Activity Theory (AT) to refine these processes and have met with some success. To date, this approach has been more useful explaining the processes post hoc. This positional paper proposes an AT method for requirement elicitation and specification definition. The method is sufficiently prescriptive and well formed that it does not require any detailed understanding of AT.

Keywords

eliciting, highly, specifying, interactive, systems, activity, theory, requirements

Disciplines

Engineering | Science and Technology Studies

Publication Details

Brown, R. B. K., Hyland, P. & Piper, I. C. (2005). Eliciting and specifying requirements for highly interactive systems using activity theory. Integrating Software Engineering and Usability Engineering (INTERACT 2005)

Eliciting and Specifying Requirements for Highly Interactive Systems using Activity Theory

Robert B.K. Brown Peter Hyland Ian C. Piper

Centre for e-Business Applications Research (CeBAR)

School of Information Technology and Computer Science, University of Wollongong

c/o SITACS Bldg3, University of Wollongong, NSW 2522 Australia

{bobbrown,phyland,ian}@uow.edu.au

Abstract

The processes of eliciting user requirements and formalising these into specifications are critical for the success of highly interactive systems. These processes are still poorly understood, partly because current methods are usually ad hoc and lack any theoretical basis. A number of researchers have used Activity Theory (AT) to refine these processes and have met with some success. To date, this approach has been more useful explaining the processes post hoc. This positional paper proposes an AT method for requirement elicitation and specification definition. The method is sufficiently prescriptive and well formed that it does not require any detailed understanding of AT.

Keywords

Requirement Elicitation, Activity Theory, Requirement Specification, Interaction Quotient

1. Introduction

Reportedly high abandonment rates and high cost overruns of computer based projects [14, 11] indicate probable misconceptualisation or misrepresentation of the purpose and requirements of the systems under development, requiring costly and time-consuming re-engineering effort. It would seem that the products under development are too often not the product that stakeholders and users actually want.

Formal methods of generating and specifying requirements have a chequered past when it comes to dealing with interface design. The Human Computer Interface (HCI) community have not adopted formal methods with open arms [12].

If the accurate determination of stakeholder requirements is a significant factor in determining software project success, then perhaps we can turn to a theory based in psychology and sociology to understand these requirements. We propose Activity Theory as a

basis for system design method. It is anticipated that such a method could bring significant benefits, especially to highly interactive systems.

2. The Interaction Quotient

Increasing numbers of systems seemingly exist almost purely for the direct use of users and cease to function in any meaningful way without user interaction. A web-based e-commerce system would exemplify such a system.

We adopt the term Interaction Quotient (IQ) [2], which reflects the proportion of a system's functionality which directly interacts with the user. Much, if not most, of the functionality 'resides' in relatively close proximity to the user interface(s) of high-IQ systems.

A greater number of high-IQ systems are being encountered over time as the World Wide Web becomes the medium for widespread access to systems; particularly those designed for use by a heterogeneous audience. A market for high-IQ systems exists in which we cannot assume any users' prior computer skills. Ease of use is therefore a crucial factor.

Defining the Requirement Specifications for high-IQ systems necessarily involves paying close attention to the so-called Non-Functional Requirements (NFR's) often associated with the user interface.

It is one claim of this proposal, that for high-IQ systems, a number of aspects traditionally considered to be NFR's are in fact Functional Requirements (FR's). We suggest a simple alternative taxonomy of Internal and External requirements, indicating requirements for those elements visible to the User (external), and those which are not (internal).

3. Activity Theory (AT)

Exploring the construction of human consciousness, Vygotsky [18] proposed that humans conceive actions upon an internalised plane of reality. The actions of the human actor (*Subject*) upon objective reality occur via various

mediating tools (both physical and psychological). The current conception of this is shown in Figure 1.

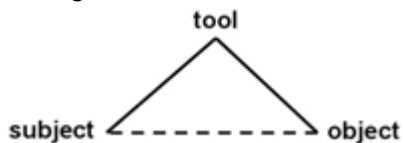


Figure 1: Vygotsky's AT conception [18]

Leont'ev [7] later focussed upon specific activities and proposed a hierarchic model shown in Figure 2, in which any given *Activity* has a Motive. Within that Motive are Goals oriented *Actions*. At the base level, atomic *Operations* are taken depending upon prevailing Conditions. Leont'ev's conception was a powerful and dynamic vision which encompassed the notion of components 'sliding' to another level (typically upwards) as the *Subject* devotes more cognitive attention upon them in the face of some unforeseen complication. Downward 'slides' may arise as *Subjects* become more familiar with *Actions* and they devolve into near autonomic operations.

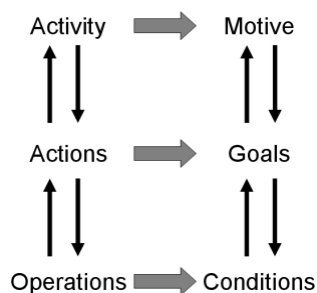


Figure 2: Leont'ev's AT hierarchy [7]

Interest turned more to the role of the people engaged in the *Activity*, and Yrjo Engström [4] developed a larger conceptual matrix, shown in Figure 3, which expands on Vygotsky's earlier work. The *Subject* is the person or sub-group whose point of view is analysed. The relationships between these socio-cultural nodes are defined in the *Division of Labour* node and also in the *Rules* node which contains social norms, regulations, domain-specific procedures and other constraints.

Engström observed that Activities are often interrelated and may impact upon one another. Kuutti [6] extended this observation and described a number of frictions between them. One of the bases of Activity Theory is the observation that an Activity may produce an

outcome which is unexpected, something of a side-effect. These outcomes may form components of neighbouring Activities. For example, one Activity may yield an outcome which contributes to, or even defines, the *Rules* or the *Tools* etc. of another. [16]

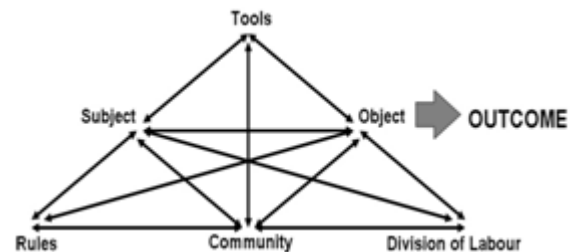


Figure 3: Engström's AT matrix [4]

Engstrom's model incorporates Vygotsky and Leontev and is now the dominant model. The reader should not feel satisfied that this brief sketch has done justice to the domain. Engström's matrix is a powerful conceptual system whose applications are widespread and much debated in a variety of fields.

Bødker's landmark PhD thesis and text "Through the Interface" [8] and later, Nardi's text "Context and Consciousness: Activity Theory and Human-Computer Interaction" [9] laid out Activity theory as a useful tool and theoretical framework for Human-Computer Interaction (HCI) study. Several proposals have come to light, notably the checklist idea [5], however it has been stated that HCI has yet to benefit directly from AT [16].

McGrath and Uden [8] observed, as have the authors of this paper, that there is a near total lack of any prescriptive procedures for applying AT in software development. They found it difficult to apply Engström and Kuutti's theoretical frameworks because prior AT case studies, were almost uniformly narrative in style and lacked a well defined and replicable process.

An exception to this lack of well defined process is the work of Vrazalic [16] who proposed a method for evaluating the usability of a system after its completion. This method method is concerned with the broader social context in which the system is used. The user, her social environment, the system and all mediating technologies must be considered. Vrazalic adopts a broader distributed definition of usability in the manner of Spinuzzi [13] that incorporates assorted genres, practices, uses and

goals. Under this notion of distributed usability, Vrazalic considers the typical usability laboratory to be an artificial environment that has a number of shortcomings that can skew the results [17]. Vrazalic's Distributed Usability Evaluation Method (DUEM) deploys a comprehensive series of tests based upon *Activity Scenarios* generated from intensive: observation of the user in their native work context, interviews with users and moderated focus group discussions.

DUEM consists of three interacting phases: understanding user activities; evaluating the role of the system in user activities and analysing and interpreting the results. The first phase produces a shared understanding of user tasks and goals. The second phase produces rich qualitative descriptions of the users' interaction with the system. The third phase concentrates on identifying points of breakdown, where the system and the activity map contradict. The problems are described via deeply contextual definitions which aid in reaching any negotiated solutions. DUEM uses the notion of distributed usability and AT principles to define contexts of a system's use by humans. Evaluation is adjudged against criteria derived from these initial findings, based upon user activity rather than system specific requirements. Users are deeply involved in an iterative process through interviews, workshops and observations. [16].

It has been observed [2] that one drawback of DUEM is that evaluators must have an understanding of AT principles to inform their analysis and to help them guide users through the process. This precludes deployment by most software analysts and requirements engineers. It is also acknowledged that it is difficult to assemble a quorum of stakeholders and developers after finalisation to conduct the method. In any event, an evaluation of a product and identification of any areas of weakness after it has been constructed can not inform its design or construction.

A chief motivation for the construction of the method briefly described in this paper is to investigate if migrating a DUEM styled AT analysis to the early-phase will address some of DUEM's identified shortfalls.

4. Why use AT?

We have chosen Activity Theory as the underlying theoretical basis for the proposed method for several reasons.

Firstly, AT has been acknowledged in the usability domain in the work of Bødker and Nardi and is recognised as a strong candidate for usability design methods. This stems from its being framed in terms of the Tool(s) used to mediate the creation of an Object by a Subject, as described above.

Secondly, the proposed method described in this paper was partially motivated in response to the AT-flavoured DUEM usability evaluation method of Vrazalic. It was hoped that by migrating the AT analysis which underlies DUEM from the testing phase to the design phase, the several of the acknowledged shortcomings of DUEM could be addressed and improvements could be achieved in the development workflow.

Thirdly, as the notion of modifying DUEM was being investigated, it became apparent that AT offered a strongly coherent and consistent terminology for describing the usability and functional requirements of the system under development. In comparison with the somewhat flexible deployment of terms under the UseCase and Scenario Planning methods, an AT informed method offers strong conceptual consistency, which meets one of the chief normative goals of the research – namely to improve on the stability and specificity of language in the process.

By way of example, at least some commonly deployed uses of the term *Scenario* in Usecase planning are functionally equivalent to the term *Activity* in AT, suggesting that a more complete theoretical conception of human Activity has much to offer.

In a further example of the suitability of AT to a design tool for highly interactive systems, we observe that Leontiev's hierarchy gives the title Operation to the most basic, atomic task, performed under some Condition – such as depressing the brake pedal in a car when a child runs in front of you. Operational tasks and their informing Conditions therefore correlate strongly with the individual buttons, switches, dialogues and component widgets of computer system interface. Likewise, we find that Leontiev's higher level, that of Goal driven Actions, is highly suggestive of how screen components may be composed meaningfully together into interface screens. The Motive driven Activity itself, Leontiev's highest level, corresponds strongly with the set of interface screens used in a given role to facilitate a specific Activity.

Fourthly, AT presents as a strong candidate for informing an end-to-end methodology. During early phase investigation it was hoped

that AT could drive the production of descriptive Requirements from user utterances. It has become apparent however that more prescriptive Specifications seem to fall out of the process with little extra work. When tied with a DUEM style cross-check in the end-phase, an end-to-end methodology seems to become an achievable goal. This paper presents early work towards that goal.

Finally, we recognise that support for reusability would be an admirable trait in a system design method. It is apparent that at least some degree of reusability may be achieved if the conceptions of the system may be decomposed and recomposed in an efficient way to cater for the reallocation of tasks and actions among the users of a system (such as when an organisation undergoes a restructure of personnel). Considering this issue, we initially investigated some of the Actor Oriented Conceptual Modelling (AOCM) proposals such as *i** [ref-Yu]. When considering that under organisational restructuring some tasks are re-instantiated and reconfigured under different coincidences of people in roles; perhaps it is better to consider the goal-driven tasks people are engaged in rather than those people who are assigned to those tasks under any single organisational structure. Accordingly, we lean towards an Activity Oriented rather than Actor Oriented conception of the system. Once again, AT presents as a strong candidate theoretical basis.

The issue of Activity Oriented Conceptual Modelling and its contribution to reusability is a strong preliminary result, and one which will be explored in detail in later papers.

5. Towards Activity Theoretical Requirements Elicitation and Specification

Although AT seems useful in requirements gathering, especially for interactive systems, a formalised and systematic method is required. This paper sets out early work in proposing such a method.

While Vrazlic has demonstrated the usefulness of AT in the evaluation of interactive systems, any Usability Evaluation Method can, at best, indicate quality “after the fact”. What is needed is a simple, yet systematic and rigorous Activity Theory method for requirements elicitation and analysis. Such a method should be sufficiently defined that the user does not need to

understand AT to use this method. Moreover, the output of this method should be a set of requirement specifications that would fit seamlessly into more conventional systems development paradigms. Thus the proposed method provides a “grey-box” set of tools that can be plugged in to the requirements elicitation and analysis process by any competent analyst, and produces specifications that are useful and understandable to any competent programmer.

It is our hope that an Activity Theoretical method for requirements elicitation and analysis should yield reliable and verifiable Requirement Specifications whilst taking account of elicited user activities. Using a common taxonomy of system-transparency, we aim to provide “grey box” specifications for the system components to be designed. We hope to present this description in the form of commonly encountered Requirements Specifications, familiar to most system builders; and we further hope that the method will be clearly prescribed as to allow analysts without AT experience to use it, most especially for high-IQ projects.

The preliminary conception for the complete method comprises four components:

- (1) an extended set of elicitation questions, laid out in a systematic and hierarchic fashion;
- (2) a template for recording the higher-level goals and objectives onto an Engström matrix;
- (3) a template for recording the lower-level goals and objectives, using a graphical conceptual matrix for the Action level, and
- (4) a set of ordered steps for processing the output and producing workable, reliable and verifiable Requirements Specification, of a form familiar to system developers.

We present some preliminary ideas for steps (1) through (3) below.

5.1. Initial Analysis

A nested sequence of elicitation questions will be put to stakeholders, clients and users.

The objective of the questions is to identify:

- ? Subjects (including sub-groups) – being those persons and groups who perform an Activity
 - o Relevant communities they belong to
 - o The Divisions of Labour between the community members
 - o Rules of membership and responsibility

- ? Identification of all Community members, which may identify Subjects of other Activities.
- ? The high level Object and Motives of these Subjects
- ? Lower level Goals within these Objects, which indicate Actions
- ? The Conditions under which Actions occur, which identifies Operations and indicate functions and controls which may need to appear as user interface components.

Note:

All the actors who share an identical point of view on the Object comprise the Subject of an Activity. Conversely, associated actors who share the Object but may perform slightly different roles may appear in the Community node.

Observe that each person (who is later fitted into a Subject group) may use different terms to describe their Activity, complicating the capture of such details [15]. The Activity Theoretic approach could and should generate an Engström matrix for *each* identified Subject, however, a layer of processing will be required to synthesize the varied Subject generated descriptions into a maximally consistent subset of Activities. The most basic Activity Theoretical principle to inform this process is that Subject-specific matrices with a common Object should merge to a single Activity, whilst those with identifiably different Objects should form distinctly separate though neighbouring Activities.

A fully developed process of this kind will elicit sufficient data to produce a set of annotated Engström matrices, awaiting processing. We anticipate that this stage of the process could benefit from a semi-automated aid, which would assist analysts who lacked significant experience in Activity Theory.

5.2 Processing the Output

From the stakeholder descriptions, a complex set of interrelated Activities – an Activity Network - will be identified. From this Activity Network, a subset of Activities will be identified as being supportable by computerised Tools, which together form our interactive system. The remaining steps in our proposed method will produce requirement specifications for these tools and for the system as a whole.

This is the first analytical scratch-board used, the System Space Build.

Each Activity in the network will have a set of Actions, elicited during the initial analysis. These Actions will be collected into a Combined Action Table (CAT), the second analytical scratch-board. Analysis of the CAT will reveal connective flows of data between the actors and their Activities. Other connective relations may be observed, such as the granting of access, a request for action, the imposing of an obligation and so on.

A resulting analytical scratch-board is the Patch Panel, which sets out all the connective relations identified from the CAT. The termini of these connective relations indicate where some interaction between users or between a user and the system occurs. These strongly correlate with interface components (Switches), which are also informed directly from Operations, as identified from Activities in the initial analysis.

The final analytical scratch-board is actually a model of the overall system. It is a hierarchic representation referred to as the “5-S” model. This model is not one of the processing steps as such, but is constructed as a part of the output.

Ultimately the method will produce as output a series of requirement specifications. Currently, we envision these having a natural-language format of a sort familiar to most system designers. Rather than following the common taxonomy of Functional, Non-Functional, however, we observe that under this analysis many of the interface related specifications usually considered NFR’s may now be seen to be functional (especially for high-IQ systems). We therefore propose a replacement taxonomy: External Requirements, being those visible to the user and comprised largely from the classifications and composition of connective relations and their Switches; and Internal Requirements, being those not visible to the user and comprised largely from the linkages between Activities.

5.2.1 The “5-S” Model

Using our AT approach to requirements elicitation and analysis, it appears that five distinct layers of specification components are supportable and useful. As shown in Figure 5, these are:

1. System.

This is a broad descriptive specification of the common computerised system of tool(s) that best mediate and facilitate the activity(s) under examination. Care should be taken not to simply replicate the functionality of any pre-existing tools, especially computer-based ones. Consider also, that it is most unlikely that all Actions within the Activity will require the mediation of the system; telephones, paper, physical aids and face to face conversation among many other tools, are likely never to be fully subsumed into computerised systems. Initially we envision the System as a space whose external boundary only is known. The specifications produced ultimately by the method should provide meaningful definition to the boundaries of this space as well as a number of insights into its internal structures by way of describing data flows and other connective relations. In this manner, we hope to produce grey-box specifications of the system.

2. Station.

The Activity Network comprises a number of Activities. Each Activity has a Subject and an Object and a comprehensive analysis of the Activity Network will identify all the discrete Activities i.e. those which do not have the same Subject and Object as any other Activity. Activities which share common Subjects may be considered to belong to a set of Roles, referred to as a Station, played by any given individual or group of individuals, which in AT terms constitutes the Subject. This is a reflection of the stakeholders and the structure of their activities which are under investigation. Note that similar Activity Networks may exist with differently composed Stations. For example, two different client organisations may have seemingly identical objectives, high-level goals and motives and thus conduct seemingly identical Activities; however, they may assign these roles differently among their members, and such a different composition of Actions and Operations would require differently composed Stations. This is a key finding and permits the re-use of the hierarchically modular components identified by our method.

3. ScreenSet

These are associated one to one with each Activity of the Activity Network. These are sets of user-interfaces required for any one Activity. Stations may contain several ScreenSets, one for each Activity it contains.

4. Screen.

Each Screen should correspond closely with one or more identified Actions of the Activity which

informs its parent ScreenSet. Identification of the Screen emerges from careful analysis of the previously observed connective relations. Any ScreenSet may exhibit a functional aggregation of such relations, which will suggest the formulation of a single interface Screen. In this manner, our method provides some insight into a workable composition of interface components to accord with the Activities of the stakeholder(s).

5. Switch.

AT defines an Operation as a simple, even autonomic response to changing conditions. In our method, each Operation may correspond to a Switch i.e. some component on the screen. These Switches may take the form of a menu, a single menu item, dial, button, dialogue, slider or any other mechanism for low-level human-computer interaction.

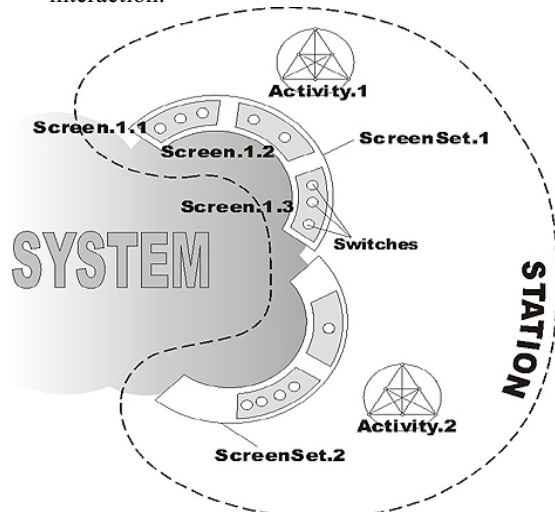


Figure 5: The "5-S" Model

Note:

As described above, according to Leont'ev, elements of the Activity residing at one level, may occur at a higher level, often under error-related circumstances. To some degree this is subsumed in the human use of the system; when a user encounters some difficulty with an Action, they may seek help or deploy other tools or rules to aid them, and thus elevate an Action to the status of an Activity.

At a more concrete level, this hierarchic upwards slippage is accommodated by the notion that a given Operations level component (Switch) may open up an Action level component (Screen) to resolve an issue or allow for configuration etc. A Screen level issue may

give rise to the need to step into, or summon, another Role level component (ScreenSet). Hierarchic ‘slippage’ downwards, typically familiarity-driven, arises when for example, an Action becomes autonomic. A well-designed system should identify such potential down-slips and provide the user with the means to assign an entire Action to a Switch, in the manner of a macro or customised button. Such notions are already well known in HCI and need no further explication here. For further detail we recommend the work of Jacob Nielsen. [10]

We observe that in building these five layers of specification, the System, ScreenSets and Switches should be more-or-less directly derivable from the Activity analysis. Deriving Stations and Screens however requires making informed choices as to the clients’ organisational structure of roles and the composition of connective relations and Switches from the Patch Panel.

5.2.2 Patch Panel

Defining the layers of the specification is not sufficient in itself to provide workable speculations. At this stage we have merely demonstrated the shape of the System and sketched the facets with which users interact (Stations, Screens etc). To yield a true grey-box specification, it is necessary to identify some of the inner workings behind the interface. [1]

In this early proposal, these take the form of connective relations primarily between both Screens and Stations. In a sense, these may be thought of as the ‘wiring’ behind and between the ‘control panels’ of the grey box.

Indeed, we choose to view the ScreenSets, Screens, Switches and connective relations as analogous to a patch-board. The CAT will indicate the locations of the ‘wiring’ junctions, but it is the composition of these ‘wires’ which will define individual Screens.

There are three principle classes of connective relations we need to identify:

1. Intra-Screen

These join switches to switches within one interface, and yield External requirements of the interface layout and design.

2. Intra-Screenset

These join between interface elements from Screens within the one ScreenSet, and thus indicate the need to move from one interface to another. Our intuition is that the composition of

Screens should attempt to minimise the number of such moves.

3. Inter-ScreenSet

a. Station-to-Station

These represent linkages between Actors and directly reflect the passing of data, control, responsibility etc between users.

b. Intra-Station

These join screen elements from screens within the one ScreenSet, and thus indicate situations where an actor needs to change ‘mode’ as they adopt one of their other ‘roles’.

Within these classes of connective relations, we hope to be able to identify and classify ‘joins’ of data flow, of access, of control, of responsibility, issues of compatibility and other aspects relevant to the framing of Requirements Specifications for the System. We anticipate greater formality may be seen if connective relations are expressed in terms of Deontic and Temporal logics.

6. Sample Activity Network

To illustrate the proposed method, we consider an activity familiar to many academics; that of administering an undergraduate assessment task. Here an Academic (A) devises an assignment task and a marking-scheme. A Tutor (T) employed under (A) deals most directly with the Student (S). The student receives the assignment from the Academic, performs the requested task, and submits a paper back to the Tutor, who marks it, according to the marking-scheme and forwards the results to the Academic for submission into a permanent record.

Upon examination of this Activity, reflection on our own experiences and casual interviews with our peers, tutors and students, we derived tables of Actions. We carefully avoided recording the functions of any of the extant computer-based systems that are currently used at our test site.

There are numerous Activities in this network, one for each Objective in accordance with Activity Theory. These may readily be clustered within three Stations (Roles), those of Academic (A), Tutor (T) and Students (S). This paper proposes the approach only, so for the sake of simplicity and brevity, this example subsumes ScreenSets within their parent Stations. Likewise, we do not here present the individual

Activity Tables, the System Space Build or the Patch Panel. A partial CAT is here given, with brief indications of some components that derive from it, simply to offer the flavour of the method. A more complete working of this example will be given in future publications.

From the partial CAT below, Entry 3 indicates a data flow from (A) to both (S) and (T). This implies the presence of 'send' and 'get' Switches. Entry 4 however requires secure transfer, implying an internal requirement of encryption and/or access control and external requirements for appropriate Switches.

Entry 5 implies a conversational exchange which may benefit from properties of simultaneity and/or persistence. A simple email module may not suffice.

Entry 16 implies a secure long-term storage facility to which (A) has write access and (S) has limited read only access. Deeper analysis of the connective relations at (A) may suggest that the Entry 16 Switches reside in a different Screen (or ScreenSet) to, say, the Entry 3 Switches.

No	Screen Sets	Actions	Comment
...			
3	(A)(S)(T)	(A) Send assignment > (S),(T) (T) Get assignment < (A) (S) Get assignment > (A)	document transfer
4	(A)(T)	(A) Send mark-scheme > (T) (T) Get mark-scheme < (A)	document transfer SECURE
5	(A)(T)	(T) Query <> (A) (A) Field query <> (T)	exchange SECURE
6	(T)(S)	(S) Query assignment <> (T) (T) Field query <> (S)	exchange
...			
11	(T)(A)	(T) Submit mark > (A) (A) Get mark < (T)	document transfer SECURE
12	(A)(T)	(A) Check mark <> (T)	
13	(A)	(A) Adjust mark	create local record
...			
16	(A)(S)	(A) Declare mark > (S) (S) Get mark < (S)	write-read secure
...			

Table 1: Extract of a Combined Action Table (CAT) for the Activity Network, 'administer undergraduate assignment'

7. Future Work

Several foreshadowed papers will set out each of the method components in greater detail within a fully worked case study; these being the Initial Analysis, the System Space Build, the

Combined Action Table, the Patch Panel and the 5-S hierarchic model.

We see potential for the Initial Analysis to be facilitated by use of a semi automated tool. The nature of such a tool will be informed by the full development and explanation of these steps.

We anticipate that a useful degree of formalisation may be introduced to the formulation of Requirements by deploying elements of Deontic and Temporal logic notations to the analysis of the Patch Panel.

As observed in the brief description above, the Station level of the model can be seen to reflect organisational structure(s) and their analysis may permit the re-use of the hierarchically modular components identified by our method. This is, in itself, a significant result and will form the substance of a later paper.

8. Conclusions

Our method shows potential to be a systematic and prescribed process with a solid theoretical base. We believe it will elicit useful Requirements from statements elicited from stakeholders without requiring the analyst to have a deep knowledge of Activity Theory.

For high-IQ systems at least, we note that the NFR/FR taxonomy commonly applied to issues of usability and interface design, has less meaning under our representation. We therefore propose the simple External/Internal requirement taxonomy, tied directly to the analysis of a given Systems' Patch Panel.

We believe our system has the potential to address an ongoing fundamental problem in system analysis and design, that of the coincidence of roles in users. Many systems are designed with a role-based interface division unmapped to organisational division of labour, and incapable of ready translation or adaptation in the face of any organisational re-distribution of roles.

9. References

- [1] Arpaci-Dusseau, A.C. & R.H., "Information and Control in Gray-Box Systems", *proceedings of the 18th ACM Symposium on Operating System Principle Vol35 Issue 5*, 2001
- [2] Brown, R.B.K., "Design-Side Considerations: A Reaction to DUEM", *proceedings of the Australian Conference on Computer Human Interaction, (OZCHI'04)*, Wollongong, 2004

- [3] Bødker, S., *Through the Interface: A human activity approach to user interface design*. Lawrence Erlbaum Associates, Hillsdale, NJ. 1991
- [4] Engström, Y., *Learning by Expanding: an activity-theoretical approach to developmental research*, Orienta-Konsultit Oy, Helsinki, Finland, 1987
- [5] Kaptelinin, V., Nardi, B., & Macauley, C., *The Activity Checklist: A tool for respecting the "space" of context*, interactions, Vol ^ Issue 4, pp.27-39 1999
- [6] Kuutti, K., *Activity Theory and its applications to information systems research and development*, in Nissen, H.E., Klein, H.K., & Hirsheims, R. (Eds) *Information Systems Research: Contemporary Approaches and Emergent Traditions* Elsevier Science, Amsterdam, pp.529-549, 1991
- [7] Leont'ev, A.N., *Activity, Consciousness, and Personality*, Prentice Hall, 1978
- [8] McGrath, G.M. & Uden, L., "Beg, Borrow or Steal: OK, but it's not all One-Way Traffic!", 2000, <http://citeseer.ist.psu.edu/668644.html>
- [9] Nardi, B. A., *Context and Consciousness: Activity Theory and Human-Computer Interaction*. MIT Press, Cambridge, MA, 1996
- [10] Nielsen, J., *Usability Engineering*, Academic Press, Boston, 1993
- [11] OASIS Study, "Why do IT projects so often fail", *OR Newsletter*, 1996, 309, pp12-16.
- [12] Paterno, F., & Palanque, P., "Formal Methods in Computer Human Interaction: Comparisons, benefits, Open Questions", *workshop session, CHI'96*, 1996
- [13] Spinuzzi, C., "Grappling with distributed usability: A cultural-historical examination of documentation genres over four decades". *Proceedings of the 17th annual international conference on Computer Documentation*, 12-14 Sept. 1999, New Orleans, USA.
- [14] Standish Group, The CHAOS Report, 1994, http://www.standishgroup.com/sample_research/chaos_1994_1.php
- [15] Sommerville, I., P. Sawyer, and S. Viller, *Viewpoints for requirements elicitation: a practical approach*, Proceedings of the 3rd IEEE international conference on requirements engineering (ICRE'98), Colorado Springs, USA, 6-10 Apr. 1998, pp.74 – 81.
- [16] Vrazalic, L., "Towards Holistic Human-Computer Interaction Evaluation", PhD Thesis, University of Wollongong, 2004
- [17] Vrazalic, L., Gould, E., "Towards and Activity-based Usability Evaluation Methodology". ... 2003
- [18] Vygotsky, L.S, *Mind in Society* Harvard University Press, Cambridge, MA, 1978