

2006

# Specifying System Requirements Using the 5S Method

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: <http://aisel.aisnet.org/acis2006>

---

## Recommended Citation

Brown, Robert B.K; Hyland, Peter; and Piper, Ian C., "Specifying System Requirements Using the 5S Method" (2006). *ACIS 2006 Proceedings*. 100.

<http://aisel.aisnet.org/acis2006/100>

This material is brought to you by the Australasian (ACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ACIS 2006 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

## Specifying System Requirements Using the 5S Method

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

### Abstract

*The 5S Method takes an Activity Theory approach to stakeholder elicitation. We interview stakeholders and build up a network of Activities according to Activity Theory precepts. These Activities are hierarchically decomposed into the smaller units of Action and Operation. This comprises a useful conceptual map of the human motivations and dependencies prior to insertion of the new computer based system. To identify and describe such a potential computer system, we analyse and identify linkages between the Activities and assemble a list of those which could usefully pass through such a computer system. Design decisions are driven and evaluated by reference to the conceptual map. We believe a description of these linkages and their relations forms an effective requirements specification for the computer system. We contend that there are advantages for conducting the whole software development lifecycle under a unifying and consistent theoretical framework. In this paper, we briefly illustrate how a System Requirements Specification might be drawn under the 5S method.*

### Keywords

Requirements Analysis, System Analysis and Design, Software Architecture, User Modelling, Activity Theory

### INTRODUCTION

Initial research investigated the use of Activity Theory (AT) as the basis for an end-to-end method for the analysis and design of highly interactive, multi-user interfaces. During the construction and initial testing of the 5S method, it became apparent that it was effective in eliciting, analysing and documenting System Requirement Specifications. Importantly, our adaption of AT defeats an impasse which previously hindered AT based design. We transfer the effects of the impasse in a way that enhances re-usability and re-factoring of the design and of system components.

Whilst earlier our publications (Brown, Hyland & Piper, 2005; Brown, Hyland & Piper, 20056a; Brown, Hyland & Piper, 2006b; Brown, Hyland & Piper, 20056c) outlined the theoretical basis of the method and the initial elicitation method, this paper will show how system requirements may be specified in a test case.

### SCOPE

Because AT provides a mechanism for describing networks of directed human activity, we believe it is useful in understanding those systems that have many users, with multiple Rôles, whose activities are highly interrelated e.g. most organisational information systems. Accordingly, the scope of our research is to develop an AT-based analysis and design method specifically for highly interactive, multi-user, information systems.

The concept of an Activity Network and the task decomposition inherent in AT i.e. Activity > Action > Operation, allows the proposed method to focus on many different levels of the interaction process. At the higher levels, those of the Activity Network and individual Activities, the method would support more experienced designers who could draw on their own experience to provide solutions to lower level design issues. At the lower levels, those of Action and Operation, the method would guide neophyte analysts and designers to the selection of suitable widgets. While at times somewhat prescriptive, our method is also intrinsically flexible, allowing analysts and designers to select those parts of the method which are appropriate to their level of expertise.

### ADAPTING ACTIVITY THEORY

AT identifies an Activity as the smallest meaningful task carried out by a human subject. Vygotsky (1978) states that all human Activity is carried out by a Subject, using physical or psychological Tools to achieve some Object which may result in a physical Outcome. Engström (1987) described the structure of an Activity as a seven node matrix (Figure 1). Traditionally, AT is concerned with the cognitive ramifications of the differences between intended Object and resultant Outcome.

A number of Activities may reside near one another and interact, forming a network that describes a larger Process (Kuutti, 1991). The Outcome of one Activity may constitute (among other things) a Tool in another (Vrazalic, 2004). We are specifically interested in Outcome-Tool transactional and transformational relations of this kind.

To adapt AT to a system design rôle, it is necessary to shift focus away from the psychosocial and cognitive aspects of any Object-Outcome discrepancy, to an investigation of the facilitating Tool(s) of an Activity, as these could include the computer system. It is also necessary to view human Processes as networks of Activities. To construct a system which facilitates such a Process, it is necessary to understand the linkages between its constituent Activities.

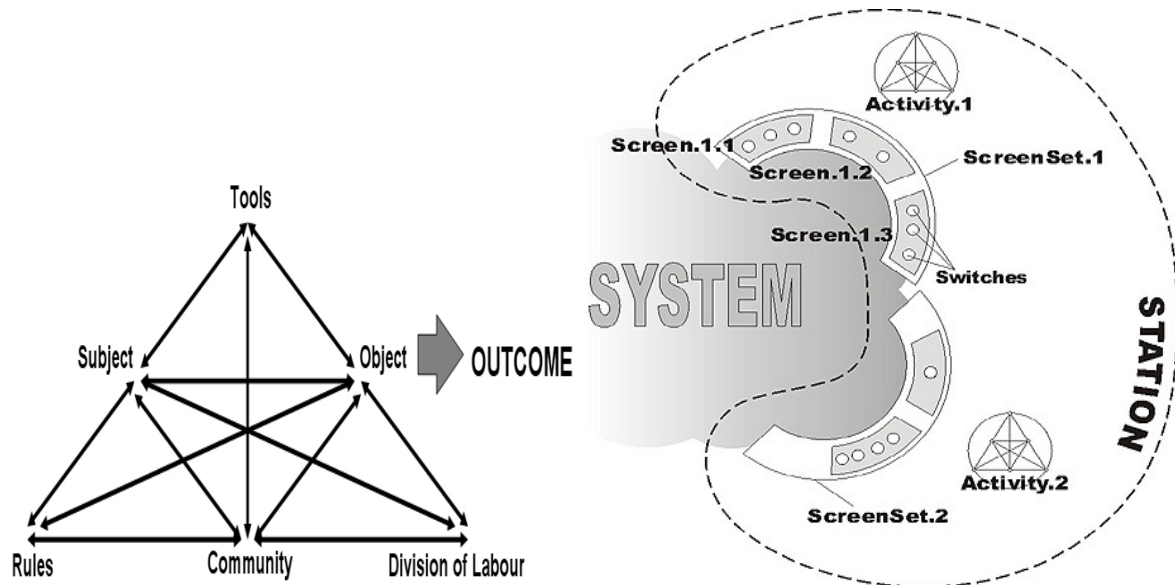


Figure 1: Engström's seven-node Activity Matrix (Engström 1987)

Figure 2: The 5S Model

Ultimately, the system may be specified by describing those Outcome-Tool transactions and transformations which may between Activities. The Designer must identify and describe them. These descriptions specify the requirements for functions to facilitate these transactions and transformations. We hope to isolate those which could usefully pass through some facilitating computer system, and we will use the term 'Instrument' to refer to data-artefacts that are passed in such transactions.

Leont'ev (1978) proposed a three layer hierarchic structure: Activity, Action and Operation to represent different levels of intellectual 'engagement' of the Subject, with an Activity requiring deep engagement while an Operation is virtually autonomic. Kuutti (1991) included fourth and topmost abstraction: the Activity Network.

We have further adapted AT to include the notion of Rôles. Any individual within the Group may be the Subject (owner) of a number of Activities, and we define a Rôle as that cluster of Activities 'owned' by an individual. These clusters of Activities with their various functions and associated UI elements are conflated to a Station, within the 5S model, as shown in Figure 2. Should a group structurally reorganise its workforce, there may be a reconfiguration of Rôles, as duties are redistributed. In such cases, 5S allows for rapid re-factoring and re-engineering by simply re-clustering the functions and UI elements accordingly.

English lacks a common collective noun for the abstract notion of 'verb', so we employ an atypical definition of 'Doing' in the singular(OED, n.d.). The collective terms 'Facilitator', 'Driver', 'Product' and 'Protagonist' were adopted for other AT aspects. We collect all the terms of our Taxonomy in Table 1.

AT layer	Doing	Facilitator	Driver	Product	Protagonist
4	Activity Network	System	Agenda	Process	Group
3	Activity	Tool/ScreenSet	Motive/Object	Outcome	Subject
2	Action	~ Screen	Goal	Transaction	Actor
1	Operation	Switch	Condition	Change	Operator

Table 1: An Extended Activity Theory Taxonomy

As indicated in AT, human activities dynamically interact with one another. Insertion of a computer system into a human Activity Network has repercussions and reshapes much of the Process. The manner in which the computer based tools are used may then be changed, and so on. Whilst interesting, these cascading effects frustrate many Activity Theoretic attempts to describe computer systems. AT inspired design methods have, as a result, yet to be broadly accepted or adopted. Indeed, some previous Activity Theory approaches to describing computer systems in their human context (Fjuk & Smrdal, 1997) seem to have balked at this impasse and instead produced powerful and useful descriptive frameworks for analysing human-computer interactions from an abstracted anthropological or sociological viewpoint.

Our approach was to realise and accept that a computer based tool, as with any physical tool, is a technological artefact which embodies theoretical and methodological contexts as they existed when built (OED, n.d.). In effect, tools are instantiated packages of theory and practice, frozen in time. Thus computer systems will fall out of favour as human Processes change around them. Rather than being deterred by this, we believe that an AT inspired design method can enhance the match between tool and user to extend the life of the tool. We also believe that a well defined design method can facilitate re-use, redesign, re-factoring and re-engineering which may allow the tool sufficient flexibility to change with the human Process. Future publications will discuss re-use in detail.

## AN AT ANALYSIS AND DESIGN METHOD

The 5-S method elicits and decomposes stakeholder utterances, in accordance with AT principles (Brown, Hyland & Piper, 2005). Starting at layer 4, the Activity Network is identified, layer 3 then identifies Activities. Layer 2 identifies Goal driven Actions and layer 1 atomic Operations. Conditions which drive Operations are then mapped to Switches, a term we employ generically for UI elements. These are recomposed and grouped into the following UI structures (Figure 2):

- 1. System: The computer tool(s) which best facilitate the Network of Activities.
- 2. Station: Activities grouped according to Rôles within the stakeholder organisation.
- 3. ScreenSet: Group of Screens associated with an Activity
- 4. Screen: Interface groupings of Switches closely related to Actions within the Activity.
- 5. Switch: Unitary elements of the UI.

Decomposition yields a picture of the Process, expressed in AT terms. Careful analysis of the temporal and deontic constraints and requirements gathered at each layer of the permits recomposition of the Facilitators at each layer until ultimately a System (the most abstract Facilitator) is described. These constraints are recorded in the Rule and Division-of-Labour nodes of the Activity Matrix.

For each Activity, using broad AT-based questions, we elicit contents for the seven nodes of the Activity Matrix (Figure 1). Actual interviews are somewhat flexible of course, and these questions serve more as a guideline than as any kind of script. Collection and analysis of these Phase 1 indicators necessarily generates a list of strong candidate Activities, to be confirmed in Phase 2. A more complete description of these early phases appears elsewhere (Brown, Hyland & Piper, 2006b).

Figure 3 shows the method's initial workflow concept passing downwards through the AT layers from abstract to refined, and back up through the layers in recomposition. Run in this way of course, the method is assembling in AT terms, a reference model of the Process in its extant state – before any computer based system has been designed and inserted. This is not unlike assembling a Business Process Model under a number of other methods, and has the same advantages of putting forward a consensually agreed representation to facilitate clear communications between stakeholders and designers. Martins *et.al.* (1999) observed that AT precepts are fundamentally compatible with Business Process Modelling tools, such as *i\**. Simply adding an AT styled modelling method to the numerous available modelling methods adds little to the field in itself, except that the AT styled model feeds directly into our design and specification stages – described later in this paper.

Of course, running the recomposition phases five to seven immediately after the Process has been captured in a reference model does not describe any new system, as one has not yet been added to the process. The complete 5S workflow (Figure 4) requires input from the analyst-designer, who tries to identify which elements of the process could or should be best facilitated by inclusion of a computer based system. This requires some intuitive and creative input on the part of the designer, as is always true, however the AT underpinning of the 5S method provides clear guidance and heuristics, which can be justified and described directly from stakeholder utterances. The reference model is constructed in such a manner as to make designing easier, more readily justifiable and adjustable.

Observe that the workflow phases are not discrete. The boundaries of the phases are porous in both dimensions. Vertically, each of the phases tends to confirm the results of the previous, and yield candidate solutions to the next. Horizontally, there are links between the decompositional analyses at any given layer and the guidance they give to the recomposition in the upwards pass. These horizontal linkages are most important. As described elsewhere (Brown, Hyland & Piper, 2006c) 5S is able to produce useful guidelines towards the user interface (UI) design and each decision made in specifying the UI can effect system requirements and visa-versa. Not only does the 5S allow for both UI and system design to occur under one unified theoretical framework and method, for best results it requires it.

The reference model of the extant Group Process is expressed as an Activity Network, which may be represented as a graph where individual Activities are nodes and the Outcome-Tool transactions are the arcs. These relations may be captured in an adjacency matrix, which we call a Combined Activity Table (CAT).

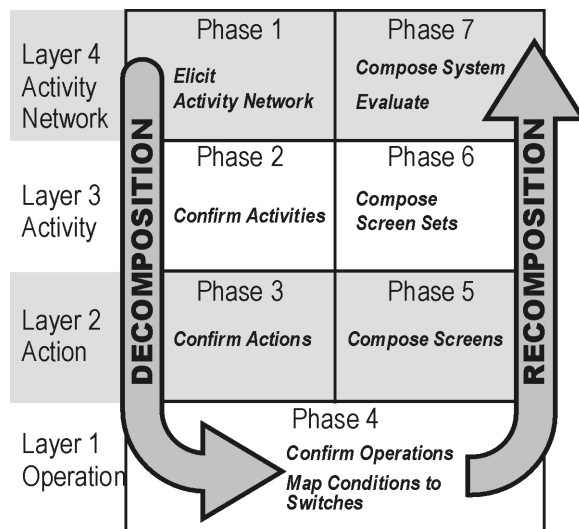


Figure 3: Initial Workflow Concept

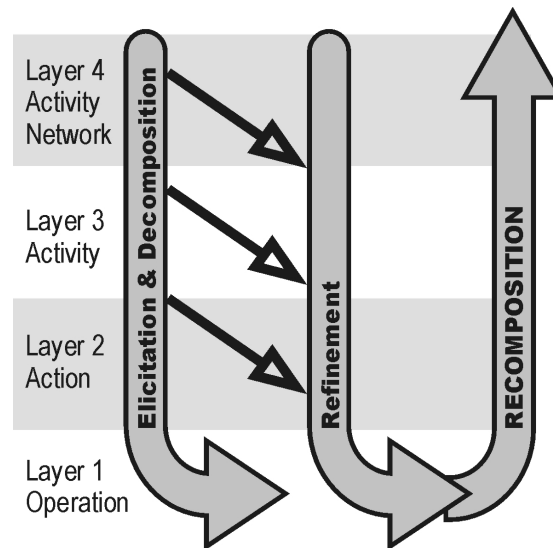


Figure 4: The 5-S Workflow

Of course, we are interested in the requirements for a system that best facilitates the Group Process. This system comprises computer based Tools which facilitate and in some instances automate some of the Group's doings. Accordingly, we are interested in those Activities whose Tools could include some element of the System. The Outcome-Tool transactions which the designer includes in the System each generate or receive an Instrument, as described earlier. All those Activities whose Tool nodes contain one or more Instruments may be imagined as sitting around the edge of some central space which contains the System itself. The System, then, is a facilitating agency for these Outcome-Tool transactions, and a medium through which the Instruments pass. Included Activities' Tool nodes sit across the boundary of the System because some of their tools reside within the System, and some with the user; the UI resides on this boundary and constitutes some further tools for that Activity. Activities whose Tool nodes do not connect to this surface are not directly considered in the design except where their constraints and requirements provide context to guide design decisions.

The analyst-designer, having built the reference model in the first decomposition pass, must now decide which Outcome-Tool transactions could pass through the System, and thus what the Instruments are. The complete 5S workflow shows this design stage as a parallel decompositional refinement pass (Figure 4). Some simple heuristics guide the designer in identifying, isolating, rationalising and even creating those Outcome-Tool transactions which are to be instantiated within the system under design. During the process of refinement, Instruments are identified and described. Reference to the various temporal and deontic constraints recorded in the reference model may indicate a need to create further Instruments. It may be the case that a proposed Instrument cannot be permitted under these same constraints. We provide indicative examples of these effects later in this paper.

The refined Process description, with a computer System now in it, is no longer purely an Activity-to-Activity network, as Instruments exist in the system itself; so Instrument transactions occur between users (Subjects) and a System container we call the System Data Repository (SDR). A CAT is no longer the most appropriate representation; instead an Instrument Table together with the hierarchic doing list serves as the System Requirement Specification.

During the process of design it is sometimes necessary to refer to 'outside', or peripheral, Activities, none of whose transactions contain Instruments, and their constraints and motivations to assist in design making decisions or, indeed, a better understanding of the overall Process. Sometimes a design decision may even draw in some

peripheral Activity to become part of the System. In cases where some Operations or Actions of an Activity can be automated, that is conducted by the System itself, the Activity may actually split or merge with another to form new Activities. This can sometimes involve incorporating a peripheral Activity, or relegating a previously included Activity to peripheral status.

Throughout the 5S workflow, it is most important that non-Instrument transactional entities remain in the analyst's documentation, even after design, as these contain vital contextual data for any future re-factoring, re-use or re-engineering of the system or any of its components. A change in the structure of work Rôles within a client organisation, for example, may require that some peripheral Activities be incorporated, or previously included Activities be taken out (if all their doings can only be conducted by humans, perhaps).

Of course, all decisions made by the designer should, we believe, involve consultation with the stakeholders. Through the reference model and the CAT, AT provides 5S with a unified and consistent taxonomy for discussion of the existing process, and of the designer's propositions.

## **TEST CASE PROCESS, "ACADEMIC ASSESSMENT TASK"**

In this Process, Academics design assessment tasks for Students to complete. Students obtain a copy of the assessment questions and, applying knowledge received in earlier classes, based on the same theory from which the assessment questions are drawn, create and submit answers. The Academic collects and marks these answers, according to the correctness of the Students' application of the received theory. The Academic must collate and centrally register all these mark results with an external administration system.

Due to space limitations, we have simplified the Process and removed the Teachers-Assistant Rôle as well as removed some Actions associated with ID verification, the appealing of mark results etc. Here we explore just the central elements of the Process – write questions, send questions, answer questions, send answers and mark answers. We omit initial elicitation and discussion of the reference model for this test case as this has been published elsewhere (Brown, Hyland & Piper, 2006b). Two user types, or Subjects, are shown here: S1 the 'Academic' and S2 the 'Students'.

The reference model for the section of the Activity Network considered in this paper is shown in Figure 5, with its CAT shown in Table 2. It indicates the following Activities:

- S1.1 Academic: Create a question document (Q) based on theory
- S1.2 Academic: Send the question document to the students
- S1.3 Academic: Create a marking guide (MG)
- S1.4 Academic: Get the submitted answers
- S1.5 Academic: Generate a report of how student answers rated against the marking guide
- S2.1 Students: Get the Question document
- S2.2 Students: Using the Questions, create an Answer document (applying received knowledge)
- S2.3 Students: Submit the Answers, and Modify as necessary, prior to deadline

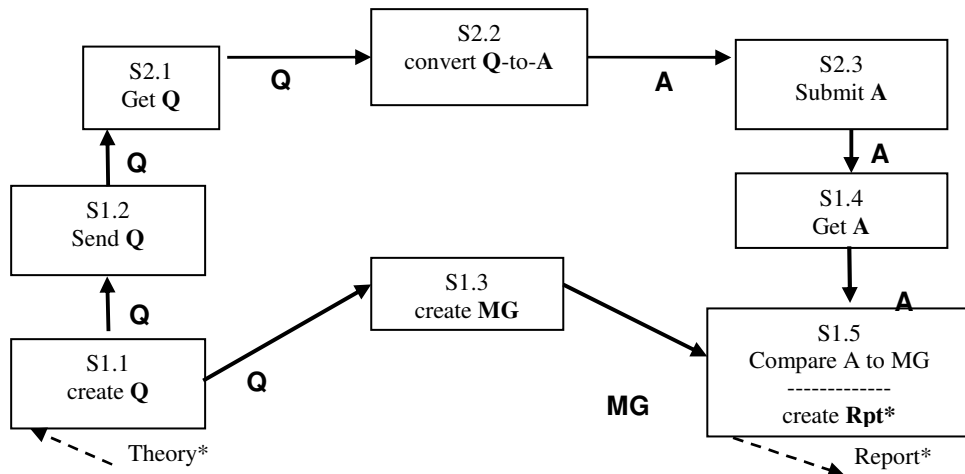


Figure 5: Reference Model of the Activity Network

ToFrom	S1.1	S1.2	S1.3	S1.4	S1.5	S2.1	S2.2	S2.3
S1.1	-							
S1.2	Q	-						
S1.3	Q		-					
S1.4				-				A
S1.5			MG	A	-			
S2.1		Q				-		
S2.2						Q	-	
S2.3							A	-

Table 2: The CAT Adjacency Matrix for the Reference Model of the Activity Matrix

## DESIGN HEURISTICS - REFINEMENT

Having assembled the reference model and its CAT, the designer now applies some simple heuristics to identify Instruments and begin to specify the System Requirements.

The designer attempts to simplify and rationalise the CAT by eliminating ‘unnecessary plumbing’ from the graph. There are two types of unnecessary plumbing: ‘Pipes’ – which are arcs (transactions) between Activities with a common Subject (owner) whereby the Subject sends something to themselves; and ‘Joints’ which are nodes (Activities) through which a data object passes unchanged.

The Pipes clearly run between the following Activities: S1.1 to S1.2, S1.1 to S1.3, S1.3 to S1.5, S1.5 to S1.4, S2.1 to S2.2 and S2.2 to S2.3. The Joint nodes are: S1.2, S1.4, S2.1 and S2.3. The strongest candidates for collapse, removal or merger are those Activities which are both a Joint and reside at one end of a Pipe arc.

Activity S1.2 “Academic sends Q” is a strong candidate. We collapse the Pipe S1.1 to S1.2 by absorbing S1.2 into S1.1. We retain the lowest Activity number, S1.1, for the remaining Activity. Likewise, we absorb Activity S1.4 “Academic gets A” into S1.5, but retain the lower number, S1.4.

Figure 6 indicates the resulting Activity Network. We have indicated the remaining Joints with dashed circles, and the remaining Pipes by shaded ovals.

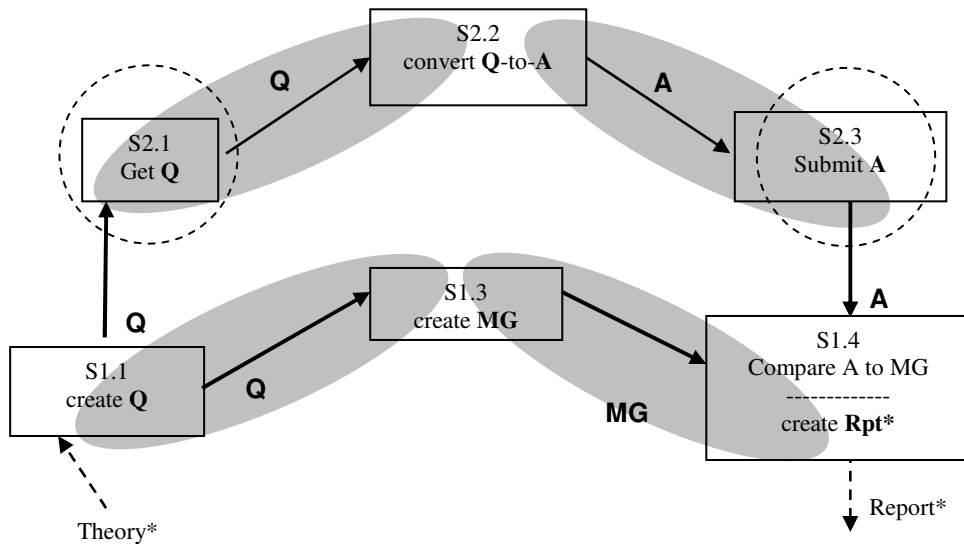


Figure 6: Partially Refined Activity Network

Staying with the Academic (S1) Activities, we see a weaker case for possible collapse. Three possibilities are presented: to collapse all three Activities into one; this is not permissible because the original reference model and the elicited constraints indicate a temporal requirement. Creating Q and MG simply must precede the comparison of MG to A. This candidate for collapse is rejected. Another candidate is to collapse S1.3 and S1.4. The elicited constraints indicate that whilst there is one MG document, there are many A documents. This indicates that one instance of MG must serve multiple ‘marking sessions’. This candidate for collapse is also rejected. The final option is to collapse S1.1 and S1.3. Creating both the questions and the marking guide seem a natural pairing – both are products of the lessons which had been previously taught to the Students. We therefore accept the collapse, and subsume S1.3 into S1.1, adopting the lower number (S1.1) for the new Activity “Academic creates and sends Q, and creates and sends MG”.

Let us briefly consider the Students (S2) Activities. According to our heuristics, Activities S2.1 and S2.3 are strong candidates for collapse. However, when we examine the elicited constraints and consider the elicited Agenda for the entire Group Process, we see that the whole purpose of these doings is to test if the Students (S2) can successfully convert Q into an acceptable A (according to the MG) by way of applying their previously received lessons. If we allow these collapses to occur, the entire process quickly collapses into a trivial case, whereby the student is simply given a 100% mark. If we consider that, at a larger scale, there is some administrative system (beyond the scope of our Activity Network) which sends enrolment data to our system, and receives a marking report in return, then we can imagine the ridiculous situation where a student is instantly awarded top marks as soon as they enrol in the subject! We therefore reject the S2 candidates for collapse.

Observe that Activity S2.2 “Student uses the Questions to create an Answer document” must occur outside the System. In fact, the processing of S2.2 necessarily occurs within S2’s own brain. The ‘data objects’ passed between S2.1 and S2.2, and between S2.2 and S2.3 cannot therefore be ‘computerised documents’ since we currently lack the technology to directly interface such objects with the human brain – the Student must ‘read Q into’ their brain, and ‘write A from’ their brain using more traditional physical and psychological tools. These data objects therefore are no longer candidate Instruments, and the Joint status of Activities S2.1 and S2.3 lapses. The case for collapse of the S2 Activities has been further weakened and remains rejected.

If an Outcome-Tool transaction between Activities does not pass through the SDR, then the data object passed by that transaction is not an Instrument, by our method’s definition. For clarity however, some non-Instrument entities such as Q\* and A\* are retained for context.



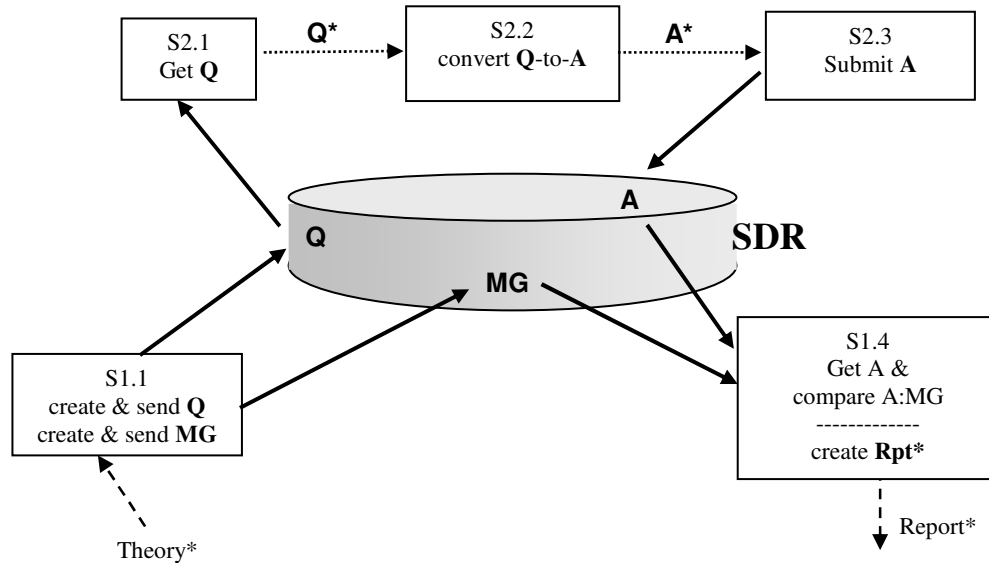


Figure 7: Partially Refined Activity Network with SDR

Figure 7 shows how these remaining Activities relate. Note that all Instruments (Q,A,MG) now reside in the System Data Repository (SDR), whilst the non-Instrument objects (A\*, Q\*, Theory\* & Report\*) do not. Here we see the first actual indications of the System itself – defined by the data objects it must contain, their natures and the temporal and deontic constraints applying to their transactions.

After applying these heuristics, consulting the stakeholders and reaching mutual satisfaction with the design, the Activity list appears as follows; the non-sequential numbering of these Activities has resulted from the refinement and design process detailed above.

- S1.1 Academic: Create and Modify a Question document
- S1.6 Academic: Create and Modify a Marking Guide
- S1.4 Academic: Create and Modify Marks for submitted student Answers & Create a final Report
- S2.1 Students: Get the Question document
- S2.2 Students: Using the Questions, create an Answer document (applying received knowledge)
- S2.3 Students: Submit the Answers, and Modify as necessary, prior to deadline

A list of this kind exists when the analyst-designer is roughly at the bottom of the ‘Refinement’ pass in the 5S Workflow (see Figure 4) and is about to process the ‘Recomposition’ pass.

Figure 8 shows Activities and Instruments after the designer has rationalised the transactions. The identified Instruments are: the question document (Q), the marking guide (MG), the Students’ answers (A) and the marks assigned by the Academic (Mks).

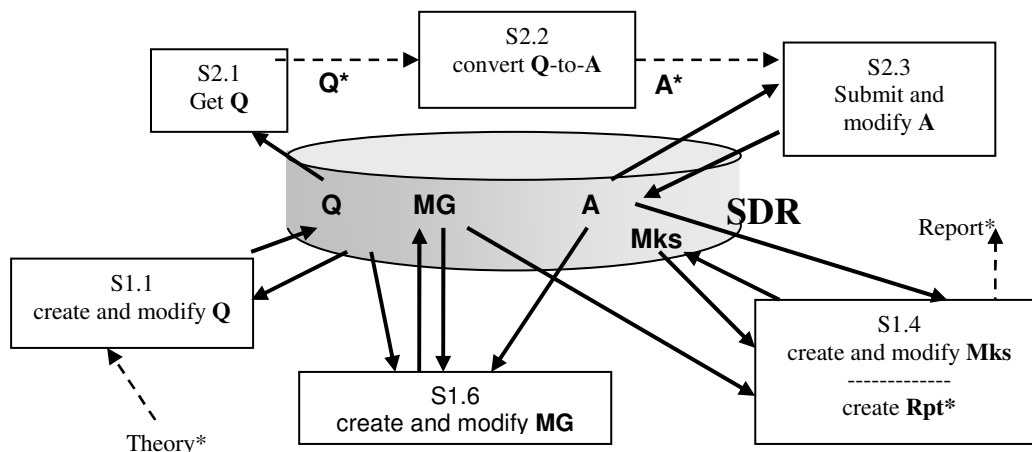


Figure 8: Initially identified system Activities, Instruments and the SDR

## BEGINNING RECOMPOSITION VIA CONSULTATION

The designer now turns to the elicited constraints and motivations, and in our opinion, should also consult with the stakeholders. From these sources, further refinement can be conducted. We present a few of these below to indicate the flavour of the method.

Under consultation, S1 (Academic) indicates a desire to modify MG, even up to the point where early answers (A) are being received from S2. This is temporally separate from the creation of Q, so we create a new Activity (with a new number) S1.6 “Academic modifies MG” which reads Q, A & MG and writes to MG.

S1 further indicates that they do not create a Report\* for each student, but one summary report. Therefore it is necessary to accumulate the scores for each A, until a Report\* can be finalised. We therefore create a new Instrument Mks (marks) which is written to and modified by S1.4 It is apparent that the Report\* cannot be generated until all answers have been marked, so we need a flag-type Instrument “Mks-Done”. It is possible for us to automate the setting of Mks-Done to the value TRUE when the number of received A’s equals the number of members of S2 (Students). S1 however does not want this to happen as soon as the final student submits, so we make the setting of Mks-Done a manual user Operation. We could create an Instrument which holds the number of members of S2 (imported from an external administration system), but S1 doesn’t deem it necessary.

S1 expresses a desire to modify the Q document, and not release it to the students until he/she is satisfied. We create a manually set Instrument “Q-Done”, which must be set TRUE before Q can be sent to the students. For the students to be aware that Q is ready to be collected there can be a number of further Instruments, but for clarity, we omit these in this paper.

S2 (Students) indicate that they wish to be able to correct and re-submit answers. This means that Activity S2.3 becomes “submit and modify A”. Our experience with the modifications of Q and of MG suggests we could create a user-settable “A-done” Instrument. The Academic (S1) however, who has higher authority over the design, vetoes allowing S2 this degree of control and opts for “A-Done” to be set TRUE when a pre-set deadline expires. From this, we easily deduce that S1 must earlier set a value for a new Instrument “DueDate”.

In previous publications (Brown, Hyland & Piper, 2006a; Brown, Hyland & Piper, 2006b) we decomposed the Activity whereby S2 ‘gets’ Q in detail. We found potential for numerous Instruments here, to facilitate validation of the student, their selection of the correct subject and assessment task, and the mode of by which Q is received. For clarity we omit these.

Activity	Instrument		
S1.1 Create & Modify Q	R/W Set Set	Q Q-done = Due-Date criteria for: auto-set A-done = TRUE	TRUE
S1.6 Create & Modify MG	R R R/W Set	Q A MG MG-done =	TRUE
S1.4 Create & Modify Mks	Req R Req R	MG-done = MG A-done =	TRUE (when Time = Due-Date)
	R R/W Set	A Mks Mks-done =	TRUE (when No.Mks = No.S2 members)
Create Rpt	Req R Create	Mks-done = Mks Rpt*	TRUE (sent to external)
S2.1 Get Q	Req R Create	Q-done = Q Q*	TRUE (sent to S2’s brain)
S2.3 Submit A	Req R/W	A* A	(got from S2’s brain)

Table 3: Instrument Table, listed by Activity

Table 3 presents the Instrument Table (listed by Activity) for the designed system after these consultative design decisions. For each Activity we list its associated Instruments. Each Activity has an operational relationship with its associated Instruments such as: read (R) the data or status of the instrument, read and/or write (R/W) the data or status of the instrument, setting (Set) the status of a flag, requiring (Req) a particular condition to be true of the Instrument, or creating (Create) an entirely new instance of an Instrument, typically for external use. An asterisk indicates the non-instrument objects passed by non-system transactions which are here included for context.

We believe that this Instrument Table, which details the objects handled by the SDR with their constraints and dependencies, can serve as the basis for a traditional Data Dictionary or similar specification document. In combination with the hierarchic Activity Network analysis, the picture 5S produces of the system allows for full specification of the UI and of the system requirements. This brief example indicates how the 5S Method can run end-to-end through the software lifecycle under a single theoretical and taxonomic framework.

## LIMITATIONS, CONCLUSIONS AND FUTURE WORK

We have identified some classes of potential Instrument which the 5S method tends to pass over. The designer should be careful to capture these. Such potential Instruments are typically associated with annotative or procedural doings: for example, the user may wish to use the system purely as a storage and retrieval facility for some non-Instrument data or to use the system itself for messaging. More infrastructural doings may relate simply to configuration of the system itself or to accessing its help documentation etc.

Failure to capture such Instruments, and to associate them with Activities, Actions and Operations (some of which are instantiated during design, purely as a result of the System's presence in the Group Process), can result in failure to identify and include all necessary UI elements in the specification.

We have not, here, concluded the refinement pass nor indicated how the UI specification emerges concurrently with the System requirements. We have also not detailed the 5S Method's strong capabilities for re-use, re-factoring and re-engineering. These will be described in future publications.

The 5S framework shows potential as a systematic method with a solid and consistent theoretical base. We believe it can elicit meaningful System Requirements from stakeholder utterances.

In this paper we have outlined in brief the principles of 5S and presented an indicative subset of results from our test case. We provided a sample Instrument Table to illustrate the general form of the system requirements 5S can produce and described how this was produced. Later publications will detail the rest of the recomposition pass and illustrate how a full data dictionary and UI specification emerges from this method.

The 5S method shows great promise as a tenable candidate for providing a theoretically consistent end-to-end software design lifecycle, which places the users' needs back in the forefront of the process.

## REFERENCES

- Brown, R.B.K., Hyland, P. & Piper, I.C., (2005) "*Eliciting and Specifying Requirements for Highly Interactive Systems using Activity Theory*" Proceedings of INTERACT'05 2005, Rome Italy
- Brown, R.B.K., Hyland, P. & Piper, I.C.,(2006a) "*5-S, an Activity Theoretic Requirements Elicitation Method for Multi-User Systems*". International Transactions on System Science and Applications, special edition. 2006
- Brown, R.B.K., Hyland, P. & Piper, I.C., (2006b) "*Extending the 5-S Activity Theoretic Requirements Method*". ICSOFT 2006, Setubal, Portugal
- Brown, R. B. K., Hyland, P. & Piper, I. C.,(2006c) "*The 5S Method: Using Activity Theory to Generate both User Interface and System Specifications* " submitted to OZCHI 2006, Sydney Australia
- Engström, Y., (1987) *Learning by Expanding: an activity-theoretical approach to developmental research*, Orienta-Konsultit Oy, Helsinki, Finland
- Fjuk, A. & Smrdal, O., (1997) "Taking Articulation Work Seriously: an activity theoretical approach" [citeseer.ist.psu.edu/fjuk97taking.html](http://citeseer.ist.psu.edu/fjuk97taking.html)
- Kuutti,K., (1991) *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
- Leont'ev, A.N., (1978) *Activity, Consciousness, and Personality*, Prentice Hall
- Martins, L.E.G., & Daltrini, B.M., (1999) "*An Approach to Software Requirements Elicitation Using Precepts from Activity Theory*" Proceedings of the 14th IEEE Conference on Automated Software Engineering ASE'99, Cocoa Beach, USA

OED (n.d.) Oxford English Dictionary Online. [www.oed.com](http://www.oed.com), Entry for "Doing, vbl.n.", 2nd definition

Pinch, T., (1986) "*Confronting Nature: The Sociology of Solar-Neutrino Detection*" Dordrecht: Reidel

Vrazalic, L., (2004) "*Towards Holistic Human-Computer Interaction Evaluation*", PhD Thesis, University of Wollongong

Vygotsky, L.S, (1978) *Mind in Society* Harvard University Press, Cambridge, MA

## **COPYRIGHT**

Robert B.K. Brown, Peter Hyland & Ian C. Piper © 2006. The authors assign to ACIS and educational and non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to ACIS to publish this document in full in the Conference Papers and Proceedings. Those documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors.