Investigating Interaction Breakdown: A Systemic-Structural Activity Approach to the Analysis of HCI Field Data

Steven Robert Harris

Hypermedia Research Unit
University of Glamorgan, Wales, UK
srharris@glam.ac.uk

Abstract

A principal aim of current research in activity-theoretical human-computer interaction (AT-HCI) is to develop practical techniques to support the design and evaluation of interactive systems. This paper demonstrates the use of an AT-based method for the design-oriented analysis of HCI field data. Morphological (structural) activity analysis is based on the Systemic-Structural Theory of Activity (SSTA), a variety of AT specifically oriented toward the study of work and learning. It involves the qualitative, algorithmic, and time-structure description of computer-mediated human activity as a structured process involving logically, spatially, and temporally organized sequences of cognitive and behavioral actions. This paper discusses the use of a human algorithm as part of the morphological analysis of an interaction sequence observed during the performance of a Web authoring task.

1 Introduction & Background

Over the last decade, AT-HCI researchers working within the tradition of the “Scandinavian School” of information technology design have established a number of methods based on general and cultural-historical activity theory (AT). These include focus-shift analysis (Bødker, 1996; Bertelsen & Bødker, 2003), the Activity Checklist (Kaptelinin et al., 1999), Activity Analysis and Development (Korpela *et al*, 2001) and the Activity Walkthrough (Bertelsen, 2003). More recently, researchers using the Systemic-Structural Theory of Activity (SSTA) have begun to develop HCI methods that offer improved support for the fine-grained description and modeling of individual and collaborative human activity during computer-mediated task performance (e.g. Bedny, Karwowski & Jeng, 2001; Bedny, Karwowski & Seglin, 2001; Bedny & Karwowski, 2003; Sengupta & Jeng, 2003; Sengupta, 2004; Harris, 2004, 2005a, 2005b). SSTA is a modern synthesis within activity theory shaped by the experience of applying AT in fields such as ergonomics, work psychology, and vocational training, and explicitly oriented toward the solution of practical design problems (Bedny, 2004, Bedny & Meister, 1997; Bedny, Seglin & Meister, 2000). SSTA understands human tool-mediated activity as a complex object of study embodying multiple, distinct aspects, and employs multiple methods to describe any single episode of activity. These methods are integrated within a recursive, multi-stage, multi-level framework and make use of standardized units of activity analysis.

Using an incident of interaction breakdown as a starting point, this paper demonstrates some techniques of morphological (structural) activity analysis through their application to HCI field data. Specifically, activity observed during the performance of a web authoring task is described through the development of a *human algorithm*, a formal symbolic description of task performance in terms of discrete goal-oriented cognitive and behavioral actions and their temporal and logical interrelationships. The methods described have been developed and applied in the context of a long-term participatory action research project involving the creative use of information technologies by non-professional people from low-income, low-education backgrounds (Harris & Shelswell, 2001; Harris, 2002, 2005b; Shelswell, 2005). To date, this research has included four phases of fieldwork. The analysis presented here uses data drawn from the first of these phases, a longitudinal field study carried out between September 2000 and May 2001 at an adult education centre in the Valleys region of south Wales, UK. In this study 27 adult basic literacy and numeracy students (14 male, 13 female aged 15-73, median age 38) were observed as they took part in 103 sessions of collaborative computer-mediated activity involving the conception, planning, execution and evaluation of individual and group digital media projects. The principal data gathering methods were participant observation and semi-structured interviews, augmented by video recording and artifact collection (described in detail in Harris 2005).

2 Method

The systemic-structural analysis of human-computer interaction activity can be carried out from the point of view of some or all of three complementary perspectives: (a) the sociocultural, which studies culture, community, and historicity, and is concerned with situating and characterizing computer-mediated activity, both historically and in relation to other activities; (b) the objectively-logical, which studies tasks, tools, work processes, and results, and is concerned with inputs, products, and the transformation process; and (c) the individual-psychological, which is concerned with actions, operations, and self-regulation, and studies subject-object relationships (Bedny & Karwowski, 2004; Harris, 2004, 2005a). Analytical techniques from the individual-psychological perspective include cognitive, parametrical, functional and morphological (structural) approaches, and involve 4 general stages: (1) qualitative description, (2) algorithmic analysis, (3) time structure analysis, and (4) quantitative (complexity) analysis (Bedny & Karwowski, 2003).

This paper presents extracts from the first two stages (i.e. the qualitative and algorithmic) of the morphological analysis of a sequence of computer-mediated activity. The main emphasis is on the objectively-logical and individual-psychological aspects of task performance. The first step in the analysis was the development of a brief description of the observed activity in objectively-logical terms, which involved determining the nature of the task (conditions, requirements, and attributes) and the objects, tools, signs, processes and results involved in its performance. Then discrete actions were isolated from the ongoing flow of activity during task performance through identification of their goal, the tools involved in their performance, and the material or symbolic object under transformation. Isolated actions were then classified according to two complementary schemes. The first differentiates types of cognitive action based on (1) the degree to which they involve direct connection with (or transformation of) the input and, (2) their dominating psychological process. The second categorizes actions according to their object (material or symbolic), and method of performance (motor or mental). The resulting catalogue of actions was then used as a basis for the generation of a symbolic representation known as a *human algorithm*, consisting of a specialized notation accompanied by explanatory text.

3 Analysis

The interaction sequence examined occurred during an observation session which took place during the period 0900-1230 on Wednesday, 4th April 2001. The main participant was a male learner, DH, aged 29. At this stage of the longitudinal study DH had previously successfully completed several digital media projects and was currently engaged in building a personal Web site. The technical means for this HCI work process were provided by an IBM-compatible, multimedia personal computer (PC) running the Microsoft Windows 98 operating system, locally networked and with broadband Internet connectivity. The PC was situated on a workbench in a small multimedia classroom, a familiar work setting for DH. Working conditions were comfortable – well-lit with a moderate temperature. DH's activity during the session involved updating two HTML (Hypertext Mark-up Language) documents - two files in ASCII format, comprising basic plain-text content marked up with HTML tags specifying formatting, hyperlinking, and other browser display properties. These were his Web site's "Links Pages"; each contained around 160 separate hyperlinks. During this session his overall task goal was to repair or remove any "dead" (dys- or non- functional) hyperlinks from the pages.

DH interacted with his files in three ways - as editable source code in the Notepad text editor, as an interactive display produced by the Internet Explorer 3 (IE3) Web browser, and as data entities manipulated using the Windows Explorer (WE) file management utility. Due to the relatively small (15 inch) CRT monitor of the PC in use, which was set at a resolution of 800 x 600 px, even with the Web browser window maximized completely scanning the contents of each Web page required DH to scroll through around 6 screens. DH's strategy for task solution was to edit the two Web pages simultaneously, making use of two instantiations of Notepad, one of WE, and multiple (1-4 at any time) browser windows. This involved switching between and refreshing windows, clicking hyperlinks and tool icons, selecting, deleting, copying and pasting text etc. For example, each time a source file was edited with Notepad, it was necessary to save the changes, and then evaluate the results in the appropriate browser window, whose display must first be refreshed. Figure 1 presents a simplified model of DH's task as a recursive, 5-step process which continues until all hyperlinks have been checked. In systemic-structural terms this step-by-step work process can be described as involving the solution of a probabilistic-algorithmic task requiring multiple decision-making actions (see Bedny & Meister, 1997, pp. 18-25; Bedny & Harris, 2005).

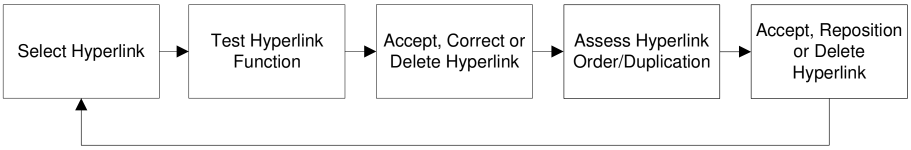
Select Hyperlink
Test Hyperlink Function
Accept, Correct or Delete Hyperlink
Assess Hyperlink Order/Duplication
Accept, Reposition or Delete Hyperlink

Figure 1: The task "update Web pages"

In order to correctly extract discrete actions from the flow of activity and develop their classification, it is necessary to identify the object, tool, and goal of each action, the precise nature of an action being dependent on the interrelation of these components in any particular set of task conditions. As a first step, a summary description of the task as a series of technological procedures is generated. This produces an idealized general description which focuses on the technical requirements. Figure 2 shows the sequence of basic technological procedures in the task "Update Web Pages". Using the sequence in Fig.2 as a starting point, actions and their operational aspects were then isolated and differentiated according to the (material or mental) object toward which they are directed, and their goal, that is, the subject's desired future result of actions toward the object. In order to isolate discrete actions it is necessary to identify the goal, object, and tools involved in *each* action, the nature of a specific action being dependent on the interrelation of these components in a particular situation.

1 Launch or restore focus to Web browser (IE3)
   a. If application already running move cursor onto window area and left click *or* move cursor onto taskbar icon and left click.
   b. If application not running, move cursor to application icon and left-click *or* move cursor to start button, left-click, navigate to appropriate menu, choose application icon or label and double left-click.

2 Load or refresh appropriate HTML document
   a. If document is already being displayed, move cursor to refresh icon (or select command from View pull-down menu) and left-click
   b. If document not displayed, load into browser by either selecting File>Open from pull-down menu, then typing file path or browsing to file location in Open dialogue box *or* use left mouse button and cursor to drag file icon from desktop or other location and drop on browser display pane by releasing mouse button
   c. In the display pane, move cursor to hyperlink being checked, left-click on link

3 View resulting display and assess

4 Make decision on whether link needs to be removed or altered

5 Launch or restore focus to text editor (Notepad)
   a. If application already running move cursor onto window area and left click or move cursor onto taskbar icon and left click.
   b. If application not running move cursor to application icon and left-click or move to start button, left-click, navigate to appropriate menu, choose application icon or label and double left-click.

6 Load or review appropriate text (.html) file

7 Identify section of markup that corresponds to link being checked

8 Cut, delete, or modify link markup

9 If moving location of link:
   a. Identify new location in appropriate text file
   b. Navigate text editor to appropriate file and location
   c. Insert cursor and paste link markup

10 Return to Step 1.

Figure 2: Sequence of basic technological procedures in the task "update Web pages"

Table 1 shows the isolation of actions involved in Step 2c of Figure 2, "in the display pane, move cursor to hyperlink being checked, left-click on link" and illustrates their classification (according to the two typologies outlined earlier). It should be noted that whereas actions 2-4 involve several external tools, no tools are defined for action 1. This indicates motor activity not involving external instruments. However, activity theory always assumes that motor actions contain cognitive components and, as such, may involve the use of internal psychological tools. For example, such tools can be assumed in action 2, which implicates not only the perception of signs visible on the interface but also their interpretation using concepts and images.

Table 1: Task isolation & classification

Action	Goal	Object	Tools	Type
1	Reach & grasp mouse	mouse		Object-practical
2	Locate display pane	Interface	Graphical interface elements	Sign-practical/ Simultaneous - perceptual
3	Move cursor over hyperlink	cursor	mouse	Object-practical
4	Activate link	hyperlink	Cursor, left mouse button	Sign-practical

4 Algorithmic Description of the Task

Algorithmic analysis generates symbolic representations known as *human algorithms*; unlike other graphical process representations such as flowcharts, such algorithms specifically depict the human elements of activity during task performance. Table 2 contains the algorithmic description of DH's activity during his performance of the task "Update Web pages". The symbolic notation denotes efferent and afferent mental and motor actions, the deterministic or probabilistic logical conditions that structure their relationships, and the various logical links between them. Each entry, or *member*, of the algorithm (one per row of the table) is accompanied by an explanatory text.

Members of an algorithm fall into two main types. *Operators* are composite units, formed of groups of 1-4 actions with their associated sub-goals, integrated through supervening task and subtask goals. They denote clusters of efferent (O epsilon) and afferent (O alpha) mental and motor actions that transform objects, energy, and information. For example, member of algorithm O^{ε}_{7} in Table 2 is an operator which includes the efferent actions (1, 3, & 4) shown in Table 1. Members of an algorithm called *logical conditions* (*l*) determine the logic of selection and realization of different members of an algorithm, and indicate the decision-making processes involved in task performance. Logical conditions may be complex (L) or simple (*l*). As can be seen in Table 2, operators and logical conditions are labeled in series with subscript numbers to facilitate reading of the algorithm. The algorithm is read from top to bottom; arrows with superscript numbers designate the logic of non-sequential (forward or backward) transitions from one algorithm member to another. More detailed descriptions of the syntax and method of development of human algorithms can be found in Bedny & Meister, (1997) and Bedny & Karwowski (2003).

Table 2: Algorithmic description of activity during performance of the task "Update Web pages"

Member of algorithm	Description of Members of Human Algorithm
$\overset{7}{\downarrow}\overset{11(1)}{\downarrow}O^{\alpha}_{1}$	Identify appropriate Web browser window
$l_{1}\overset{1}{\uparrow}$	If correct Web browser window has focus go to $O^{\alpha}_{3;}$ if not go to $O^{\varepsilon}_{2.}$
O^{ε}_{2}	Bring browser window to foreground by moving cursor onto window area and left-clicking *or* moving cursor onto taskbar icon and left-clicking.
$\overset{1}{\downarrow}O^{\alpha}_{3}$	Check to see if correct Web page is displayed
$l_{2}\overset{2}{\uparrow}$	If appropriate Web page is displayed go to O^{ε}_{4} ; if not go to O^{ε}_{5}
O^{ε}_{4}	Refresh browser display by moving mouse cursor to refresh icon and left-clicking *or* selecting command from View pull-down menu and left-clicking
$\overset{2}{\downarrow}O^{\varepsilon}_{5}$	Open correct HTML document by either selecting File→Open from pull-down menu, and browsing to file location using Open dialogue box *or* use left mouse button and cursor to drag file icon from desktop or other location and drop on browser display pane by releasing mouse button
$\overset{4}{\downarrow}\overset{4}{\downarrow}O^{\alpha}_{6}$	Identify next hyperlink to be checked

Table 2: Algorithmic description of activity during performance of the task "Update Web pages"

Member of algorithm	Description of Members of Human Algorithm
O^{ε}_{7}	Activate link by moving mouse cursor to hyperlink being checked and left-clicking
$O^{\alpha\, th}_{8}$	Look at new page display in browser window and compare with expected result
$l_3\overset{3}{\uparrow}$	If browser display is appropriate go to O^{ε}_{9} ; if "The page cannot be displayed" HTML file is displayed *or* if link works but page inappropriate go to O^{ε}_{11}
O^{ε}_{9}	Move mouse cursor to browser Back button and left-click *or* right-click in browser display pane to invoke context menu and left-click on Back item
O^{α}_{10}	Look at hyperlink and check its alphabetical order in hyperlink list display
$l_4\overset{4}{\uparrow}$	If alphabetical order is correct go to O^{α}_{6}; otherwise, go to O^{ε}_{11}
$\overset{3}{\downarrow}\overset{4}{\downarrow}\overset{7(2)}{\downarrow}\overset{11(2)}{\downarrow}O^{\alpha}_{11}$	Identify correct text editor window
O^{ε}_{12}	Bring text editor window to foreground by moving cursor onto window area and left-clicking *or* moving cursor onto taskbar icon and left-clicking
O^{α}_{13}	Check that correct HTML file is displayed in text editor by inspecting window title bar
$l_5\overset{5}{\uparrow}$	If correct file is displayed in text editor go to O^{α}_{15}; otherwise go to O^{ε}_{14}
O^{ε}_{14}	Open correct HTML file by either selecting File→Open from pull-down menu, and browsing to file location using Open dialogue box *or* use left mouse button and cursor to drag file icon from desktop or other location and drop on browser display pane by releasing mouse button
$\overset{5}{\downarrow}O^{\alpha}_{15}$	Scan HTML mark-up and plaintext in text file; identify link mark-up in text file that corresponds to incorrect hyperlink item in display
$l^{\mu\, th}_{6}\overset{6}{\uparrow}$	If alphabetical order was incorrect go to O^{ε}_{15}; otherwise go to [check mark-up]
O^{ε}_{16}	Highlight link mark-up by dragging mouse across text with left button depressed; cut mark-up to clipboard by using keystroke combination CTRL+X *or* choosing Edit→Cut from pull-down menu
$O^{\alpha\, th}_{17}$	Identify appropriate location for hyperlink mark-up
O^{ε}_{18}	Paste text from clipboard by inserting cursor in correct location and using keystroke combination CTRL+V *or* choosing Edit→Paste from pull-down menu; save file by choosing File→Save from text editor pull-down menu
O^{ε}_{19}	Bring browser window to foreground by moving cursor onto window area and left-clicking *or* moving cursor onto taskbar icon and left-clicking; Refresh browser display by moving mouse cursor to refresh icon and left-clicking *or* selecting command from View pull-down menu and left-clicking
O^{α}_{20}	Inspect updated browser display
$l_7\overset{7(1-2)}{\uparrow}$	If updated display is satisfactory, return to O^{ε}_{2} to check next link; if not, return to O^{ε}_{11}

Table 2: Algorithmic description of activity during performance of the task "Update Web pages"

Member of algorithm	**Description of Members of Human Algorithm**
O^{α}_{21}	Inspect mark-up for spelling or syntax errors
$l_8 \overset{8(1-2)}{\uparrow}$	If syntax error identified go to $O^{\mu\ th}_{23}$; if spelling error identified go to $O^{\mu\ th}_{24}$ otherwise proceed to O^{ε}_{22}
$\overset{9(3)}{\downarrow}\ \overset{10(3)}{\downarrow} O^{\varepsilon}_{22}$	Highlight link mark-up by dragging mouse across text with left button depressed; delete mark-up by using Delete key *or* choosing Edit→Delete from pull-down menu
$\omega_1 \uparrow_{\omega 1}$	Always false logical condition; default to O^{α}_1 if correction procedure completed, otherwise continue to next member of algorithm
$\overset{8(1)}{\downarrow} O^{\mu\ th}_{23}$	Identify required syntax correction
$l_9 \overset{9(1-3)}{\uparrow}$	If unable to identify required syntax correction go to O^{ε}_{25}; otherwise go to O^{ε}_{27} *or* O^{ε}_{22}
$\overset{8(2)}{\downarrow} O^{\mu\ th}_{24}$	Identify required spelling correction
$l_{10} \overset{10(1-3)}{\uparrow}$	If unable to identify required spelling correction go to O^{ε}_{25}; otherwise go to O^{ε}_{27} *or* O^{ε}_{22}
$\overset{9(1)}{\downarrow}\ \overset{10(1)}{\downarrow} O^{\varepsilon}_{25}$	Request assistance from other participants
O^{α}_{26}	Receive information on how to proceed from other participants
$\overset{9(2)}{\downarrow}\ \overset{10(2)}{\downarrow} O^{\varepsilon}_{27}$	Insert cursor at appropriate position in mark-up and make correction; save file by choosing File→Save from text editor pull-down menu
O^{ε}_{28}	Bring browser window to foreground by moving cursor onto window and left-clicking *or* moving cursor onto taskbar icon and left-clicking; refresh browser by moving mouse cursor to refresh icon and left-clicking *or* selecting command from View pull-down menu and left-clicking
O^{α}_{29}	Inspect updated browser display
$l_{11} \overset{11(1-2)}{\uparrow}$	If updated display is satisfactory, return to O^{α}_1 to check next link; if not, return to O^{ε}_{11}

5 Discussion

In order to carry out each cycle of the task "Update Web pages", DH was required to identify and foreground appropriate application windows (e.g. O^{α}_1, O^{ε}_2), inspect URLs in the source code (e.g. O^{α}_{21}) and check for syntax and spelling errors (e.g. $O^{\mu\ th}_{23}$, $O^{\mu\ th}_{24}$). These complex perceptual and thinking actions involved both short-term (when identifying correct windows and comparing code and display) and long-term memory (comparison of display with overall design plan, recalling rules of URL syntax, etc.). Inspection of the algorithm indicates that the complexity of the task became maximal at those points where DH was required to identify, highlight, cut, and then paste source code within or between Notepad windows, save the changes, and inspect the results in a refreshed browser window display, e.g. the sequence of operators and logical conditions from O^{α}_{15} to O^{α}_{20}. At these points, demands on short-term memory clearly became severe, as DH was required to store information about which code snippet was being edited, and about code content and syntax, as well as perform appropriate perceptual and motor actions with regard to both source and destination windows. In fact, it was at precisely these points in the task that breakdown was observed to occur. On several occasions during the session, DH reported being unable to correctly identify causal connections between his actions and observed results when editing operations failed to produce anticipated changes in the display. As one field note entry records:

> DH becomes confused as to which page he is actually working on, ending up simultaneously editing the same set of links embedded in two different pages. He says to me "I just can't understand why it's taking so long". (Extract from field note entry for Session 69, 4 April 2001)

After attempting to recover from breakdown by varying his strategy - e.g. taking actions such as opening additional windows - DH finally abandoned the task and sought external assistance.

Turning to the objectively-logical description of the task, it can be seen that differences in user interface design of the two main software tools in use contributed to interaction breakdown by further increasing the complexity of the task at its most critical juncture. As has been noted, during task performance DH was running both Notepad and IE3 in multiple windows. However, these two applications dealt with windows in different ways. In IE3, windows function as several views generated from the same running instantiation of the application; opening a new window is specifically defined as a menu option, and closing an open window does not necessarily exit the application. By contrast, each Notepad window represents a single instantiation of the program. With the hardware/software configuration in use, it was feasible to load several instantiations of Notepad into RAM at once without impairing performance; these instantiations acted as independent entities. This arrangement made it possible to edit the same text file simultaneously in two Notepad windows, making and saving *different* changes to the file in each of those windows. To add to this complexity, each IE3 window actually displayed not the relevant source HTML itself, but a locally cached version. Consequently, DH found himself in the situation of making changes to source files but unable to identify any corresponding change in the browser display. His understanding of the situation was not well-supported by the various elements of the desk-top display. The only visual clues available were the short filenames in the title bars of the Notepad windows - which do not include full file paths - and the complex local URLs displayed in the address bar of the browser; as has been noted, their interpretation required knowledge not only of the file structure created by the user but also of the way the application itself stored and handled temporary files.

6 Conclusion

These findings suggest that the algorithmic analysis of computer-mediated activity is capable of producing the very detailed information needed to respond to Bertelsen & Bødker's challenge of developing an activity-theoretical HCI that is both "…close to technology" and "…design oriented" (Bertelsen & Bødker 2003 p. 323). Developing human algorithms helps to identify those aspects of task performance where design intervention might prove most effective: for example, they quickly highlight instances where object, tools, or task conditions require users to perform complex, repetitive or otherwise demanding actions, and - as here - where aspects of user interface design produce problems for the user when engaged in a real-world task. When analyzing collaborative activity, individual algorithms can be constructed from the point of view of each involved user, then compared, contrasted, or combined.

It should also be noted that the development of human algorithms is only one aspect of morphological analysis. Having mapped the logical structure of task performance also makes it possible to describe the temporal structure of activity in terms of performed actions in more detail. Time measurements derived from field data can be used to specify the duration of individual elements of activity, with particular attention being paid to the structure of sequential and simultaneous performance of mental and motor actions. Such time-structure analyses can be further supplemented through the use of standardized classifications of motor actions, such as those developed within the MTM-1 system (see Bedny & Meister, 1997 pp. 252-262); they also provide a basis for the quantitative evaluation of task complexity (ibid. pp. 262-277).

This paper has focused on the analysis of HCI field data from objectively-logical and individual-psychological standpoints. However, it is clear that the structure of a person's work activity, and hence the design of means to support that work activity, is not only affected by their own individual characteristics and their physical and technical environment, but also the norms, standards, rules of behavior etc. that form the sociocultural context of their work. Therefore, in order to be fully effective in informing design, analyses such as that presented here should be used in conjunction with socio-cultural-historical descriptions of the specific communities and work practices under study. The systemic-structural theory of activity provides a consistent conceptual framework within which these differing perspectives can be effectively integrated.

7 References

Bedny, G. Z. (2004). Preface, special issue on activity theory. *Theoretical Issues in Ergonomics Science, 5*(4), 249-254.

Bedny, G. Z., & Harris, S. R. (In Press). The Systemic-Structural Theory of Activity: Applications to the Study of Human Work. *Mind, Culture and Activity, 12*(2).

Bedny, G. Z., & Karwowski, W. (2004). A functional model of human orienting activity. *Theoretical Issues in Ergonomics Science, 5*(4), 255-274.

Bedny, G. Z., & Karwowski, W. (2003). A Systemic-Structural Activity Approach to the Design of Human-Computer Interaction Tasks. *International Journal of Human-Computer Interaction, 16*(2), 235-260.

Bedny, G. Z., Karwowski, W., & Jeng, O.-J. (2001). *Activity Theory as a framework to Study Human-Computer Interaction.* Paper presented at the International Conference on Computer-Aided Ergonomics and Safety, Maui, Hawaii.

Bedny, G. Z., Karwowski, W., & Seglin, M. H. (2001). Activity Theory as a Basis for the Study and Redesign of Computer Based Task. In M. J. Smith & G. Salvendy & D. Harris & R. J. Koube (Eds.), *Usability Evaluation and Interface Design: Cognitive Engineering, Intelligent Agents and Virtual Reality* (Vol. 1, pp. 342-346). Mahwah, NJ: Lawrence Erlbaum Associates.

Bedny, G. Z., & Meister, D. (1997). *The Russian Theory of Activity: Current Applications to Design and Learning*. Mahwah, NJ: Lawrence Erlbaum Associates.

Bedny, G. Z., Seglin, M. H., & Meister, D. (2000). Activity theory: history, research and application. *Theoretical Issues in Ergonomics Science, 1*(2), 168-206.

Bertelsen, O. W. (2003). Activity Walkthrough: a cognitive walkthrough in activity theory terms. In M. Hertzum & S. Heilesen (Eds.), *Writings in Computer Science: Proceedings of the Third Danish HCI Research Symposium* (Vol. 98, pp. 17-20). Roskilde, Denmark: Roskilde University.

Bertelsen, O. W., & Bødker, S. (2003). Activity Theory. In J. M. Carroll (Ed.), *HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science* (pp. 291-234). London: Morgan Kaufmann.

Bødker, S. (1996). Applying Activity Theory to Video Analysis: How to make Sense of Video Data in Human-Computer Interaction. In B. M. Nardi (Ed.), *Context and Consciousness: Activity Theory and Human-Computer Interaction* (pp. 147-174). Cambridge, MA: MIT Press.

Harris, S. R. (2005a). Systemic-Structural Activity Analysis of HCI Video Data. In O. W. Bertelsen & M. Korpela & A. Mursu (Eds.), *Proceedings of ATIT04: 1st International Workshop on Activity Theory Based Practical Methods for IT-Design, September 2-3, 2004, Copenhagen, Denmark.* Aarhus: Aarhus University Press.

Harris, S. R. (2005b). *Design, Development, and Fluency: An Activity-Theoretical Approach to Human-Computer Interaction Research.* Unpublished PhD. Thesis, University of Glamorgan, Wales, UK.

Harris, S. R. (2004). Morphological Analysis of HCI Video Data Using Activity Theory. In A. Dearden & L. Watts (Eds.), *Proceedings of HCI2004: Design for Life* (Vol. 2, pp. 41-44). Bristol: Research Press International.

Harris, S. R. (2002). PD in Ponty: Design-by-Doing in Adult Basic Education. In T. Binder & J. Gregory & I. Wagner (Eds.), *PDC 02 Proceedings of the Participatory Design Conference* (pp. 278-283). Malmö, Sweden: CPSR.

Harris, S. R., & Shelswell, N. (2001). Building bridges across the digital divide: supporting the development of technological fluency in Adult Basic Education learners. In A. Mason (Ed.), *Proceedings of the FACE Annual Conference 2001* (pp. 42-51). Southampton: FACE.

Kaptelinin, V., Nardi, B. M., & Macaulay, C. (1999). The Activity Checklist: A Tool for Representing the "Space" of Context. *interactions, 6*(4), 27-39.

Korpela, M., Mursu, A., & Soriyan, H. A. (2001). Information Systems Development as an Activity. *Computer Supported Cooperative Work, 11*(1-2), 111-128.

Sengupta, T. (2004). *Investigating Graphical User Interface Usability on Task Sequence and Display Structure Dependencies.* Unpublished Doctoral Thesis, New Jersey Institute of Technology.

Sengupta, T., & Jeng, O-J. (2003). Activity Based Analysis for a Drawing Task, *In Proc. Ergonomics in the Digital Age, IEA Congress 2003* (Vol. 6, pp. 455-458). Seoul: The Ergonomics Society of Korea.

Shelswell, N. (2005). *A Critical Investigation into Uses of New Technologies in the Teaching and Learning of Adult Literacy.* Unpublished MPhil. Dissertation, University of Wales Institute, Cardiff.