New graphical and text-based notations for representing task decomposition hierarchies: towards improving the usability of an Ergonomics method

Huddlestone, JA & Stanton, NA

Original citation & hyperlink:
Huddlestone, JA & Stanton, NA 2016, 'New graphical and text-based notations for representing task decomposition hierarchies: towards improving the usability of an Ergonomics method' Theoretical Issues in Ergonomics Science, vol 17, no. 5-6, pp. 588-606.
https://dx.doi.org/10.1080/1463922X.2016.1201168

DOI 10.1080/1463922X.2016.1201168
ISSN 1463-922X
ESSN 1464-536X

Publisher: Taylor & Francis

This is an Accepted Manuscript of an article published by Taylor & Francis in Theoretical Issues in Ergonomics Science on 13 July 2016, available online: http://www.tandfonline.com/10.1080/1463922X.2016.1201168.

New graphical and text-based notations for representing task decomposition hierarchies: towards improving the usability of an Ergonomics method

John A. Huddlestone (corresponding author)

Centre for Mobility and Transport, Coventry University, Coventry, UK
Faculty of Engineering and Computing
Coventry University
Priory Street
Coventry CV1 5FB
United Kingdom

John.huddlestone@coventry.ac.uk

Neville A Stanton
Transportation Research Group, Civil, Maritime, Environmental Engineering and Science, Faculty of Engineering and the Environment, Bouldrewood innovation Campus, University of Southampton, Burgess Road, Southampton, UK

Dr John Huddlestone is a Senior Research Fellow in Human Factors within the Engineering and Computing Faculty at Coventry University in England. He served as commissioned officer in the Royal Air Force from 1983 until 2003 when he joined academia. A co-author of the Team and Collective Training Needs Analysis Methodology published by Ashgate in 2016, his research interests include team training, simulation, human factors methods and aviation human factors. He is currently involved in research into the human factors of future flight deck technologies and crewing concepts, and human factors methods. In 2008 he was awarded Institute of Ergonomics and Human Factors President's Medal for outstanding contribution to Human Factors Integration research, as a member of the Human Factors Integration Defence Technology Centre team. He holds a PhD in Applied Psychology from Cranfield University, a Master's degree in Computing Science from Imperial College, London and Batchelor's Degree in Education from Notting Trent University. He is a Chartered Information Technology Professional and Member of the British Computer Society.

Professor Neville Stanton, PhD, DSc, is both a Chartered Psychologist and a Chartered Engineer and holds the Chair in Human Factors in the Faculty of Engineering and the Environment at the University of Southampton. His research interests include modelling, predicting and analysing human performance in transport systems as well as designing the interfaces between humans and technology. Professor Stanton has worked on cockpit design in automobiles and aircraft over the past 25 years, on a variety of automation projects. He has published 30 books and over 240 journal papers on Ergonomics and Human Factors, and is currently an editor of the peer-reviewed journal Ergonomics. The University of Southampton have awarded him a DSc in 2014 for his sustained contribution to the development and validation of Human Factors methods.

New graphical and text-based notations for representing task decomposition hierarchies: towards improving the usability of an Ergonomics method

The representation of task decompositions, in the form of sub-goal hierarchies and their related sequencing triggering and exit conditions, lies at the heart of Hierarchical Task Analysis and related techniques. Analysis of the conventional graphical and text-based notations for these representations, using the principles of cognitive load theory, identifies a number of features that may give rise to difficulties when reading and constructing HTA representations. A revised graphical notation is presented, derived from similar notations that are used in the software engineering and human factors domains. An equivalent text-based notation is also presented to facilitate the representation of the task decomposition tabular format, where additional details of the task can be captured. It is suggested that the use of these revised notations could improve usability when constructing and interpreting graphical and tabular representations of hierarchical task decompositions.

Keywords: hierarchical task analysis, graphical notation, cognitive load, usability, ergonomics methods.

Relevance to human factors/ergonomics theory

This paper illustrates the application of human factors principles to the evaluation of the usability of a human factors methodology. Specifically, cognitive load theory is used to evaluate the usability of graphical and tabular notations used for representing task decompositions in hierarchical task analysis. Usability criteria for task analysis notations are identified and used to support the analysis. Based on the findings of the analysis, the same principles are then used to guide the development of revised graphical and tabular notations which include representation forms for recurrent operations. The properties of the revised notation are then evaluated against the identified usability criteria. The approach taken could be applied to the evaluation of the usability of the notations used in other human factors methods.

Introduction

Task analysis is a core human factors technique and is typically required in any human factors analysis effort (Stanton et al 2005, 2013). A key component of the task analysis process is the production of a documented representation (also referred to as a model) of the task that has been analysed. In the context of human computer interaction, Paris et al (2000) and van Welie et al (2000) echo the thoughts of Lim and Long (1994) in underlining the importance that these representations have in supporting communication during systems development, which they suggest is one of their major values. They cite the use of task models to support communication between a wide range of people from different backgrounds, such as software architects, interface designers, end users, and various stakeholders in activities such as validation of user requirements, definition of the vocabulary to be used in a user interface, and validation design feasibility. Annett (2004) also identifies the more general requirement to check the validity of a task analysis by inviting stakeholders to review the task representation to identify misinterpretations and omissions. Balbo et al (2004) observe that such communication requirements present a usability challenge as task representations have to be easily read by people without a background in task analysis, as well as the analysts themselves. They also note that task modelling notations are not typically designed explicitly for the purposes of communication. A content analysis of industry perspectives on task analysis by Stanton and Diaper (2004) identified difficulties faced by non-analysts in reading task analysis outputs to be a common concern.

Annett (2004) makes a poignant comment about human factors methods when he states that “it is reasonable to expect that the same standards of usability should apply to the methods used by human factors specialists as apply to the objects of their study” (p80). This paper aims to take a step towards the achievement of this expectation by applying human factors principles to the evaluation and revision of examples of task analysis notational forms, with the goal of improving their usability both for analysts producing task representations and stakeholders who may be required to read and interpret them.

The task representations of Hierarchical Task Analysis (HTA) have been selected for analysis in this paper, with Cognitive Load Theory (CLT) providing the theoretical framework for the analysis. The next section provides a brief introduction to HTA and the justification for its selection as the subject of this study. After a brief explanation of the choice of CLT as an analysis framework, examples of HTA task representations are analysed. Revised graphical and textual notations for HTA task representations are then developed to address the issues identified in the analysis. This is followed by the illustration of its use in recasting the examples used in the analysis section and an evaluation of its merits form a usability perspective.

Hierarchical Task Analysis

HTA, developed by Annett, et al (1971), is a widely used task analysis method. It has been characterised Ainsworth and Marshall (1998) as “perhaps the nearest thing to a universal TA [task analysis] technique” (p1611). Stanton (2006) notes that many HF methods require an HTA as an input or are made easier if an HTA is available. Consequently the, use of hierarchical task decomposition representations is common in the human factors domain.

Annett et al (1971) conceptualise tasks as being operations that can be defined by their goals. They can be decomposed into sub-operations, defined by sub-goals, and these are represented in a sub-goal hierarchy. Critically, the sequencing, triggering and exit conditions for sub-operations have to be identified. The description of the order in which sub-operations are carried out, and their triggering and exit conditions, is called a plan. Annett (2004) attributes the central HTA concepts of hierarchical decomposition of tasks and the use of plans to the work of Miller, Gallanter and Pribram (1960) on feedback theory. HTA employs complementary graphical and tabular task representations of sub-goal hierarchies. The graphical representation provides an overview of the task, whilst the tabular format facilitates the capture of more detail about the task components. Plans may be represented in HTA diagrams as textual annotations on the graphical representation of the sub-goal hierarchy or in a flowchart format. Much of the ubiquity of HTA is probably owed to the flexibility of the approach these complementary representations offer (Stanton, 2004).

Whilst HTA use has been widespread there have been issues identified with use of the method. Stanton and colleagues have reported that HTA is one of the most time consuming methods to train novices in (Stanton and Young, 1998, Stanton et al, 2014). Shepherd (1976) observed that only one of a group of training officers new to HTA attempted to record plans in the task decomposition. Recent studies on HTA training shed further light on the difficulties that novices experience in conducting HTA. Patrick et al (2000) found that novices experienced difficulty with hierarchical analysis, often producing representations that were more akin to flowcharts. They also report that most were unable to formally specify plans, although they note that the activity may simply have been forgotten. The correct identification of task boundaries and determination of when to stop the analysis also proved problematic. The findings of Adams et al (2013) were very similar. They report that novices having difficulty with hierarchical decomposition and the production of plans, typically only producing one, top-level plan. They also experienced difficulties with correctly identifying task boundaries. Whilst some caution is needed in generalising these findings because, as Patrick et al note, one would expect novices to make mistakes, and in each of these studies only a few hours of training were provided, it is interesting to see what types of errors persist in professional practice.

Ainsworth and Marshall (1998) conducted a survey of task analyses carried out in the defence and nuclear industry sectors in which they evaluated task analysis reports and collected additional data from the analysts where possible. Some 90 studies were considered. In the military sector reports, half included HTAs but it was found that plans were only developed in the minority of cases. HTAs were also widely used in the nuclear sector reports. By contrast, plans were included in the majority of the analyses. Where they were not, it appeared that the tasks were linear sequences. Whilst plans were mainly described using text, flow diagrams were used in some cases, but these were often unclear for complex plans. Ainsworth and Marshall describe the quality of the HTA reports as variable, with one third not including an HTA diagram and some of the HTA diagrams being very cluttered. Furthermore, users who were unfamiliar with HTA reported that they sometimes misinterpreted the HTA representations. Some caution is also needed in interpreting these findings, as they are not diagnostic as to the cause of the weaknesses exposed. However, the fact that there appears to be a consistent theme relating to the production of plans, and that consumers of the task representations as well as the producers have experienced difficulties with them, suggests that investigation into the properties of the notations used to identify if there are any features which may propagate errors in the construction and interpretation of task representations is merited. Furthermore, given the ubiquity of HTA as an ergonomics method, any developments which enhance its usability could potentially be of benefit to a wide audience in the ergonomics community.

Cognitive load theory as an analysis framework for the usability of HTA notations

Usability is defined by the International Standards Organisation (ISO) as the 'extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use' (ISO 2015:p9). Balbo et al (2004) identify two key usability criteria for task modelling notations:

Usability for communication: the ease with which a task model produced using the notation can be read and understood for the purpose of communication between the originators of the model and other stakeholders.

Usability for task modelling: the ease with which task models are generated and modified, which requires communication within the task modelling team.

They suggest that, to support these constructs, a task notation must be easy to read and quick to learn by both novices and professionals.

Cognitive Load Theory (CLT) suggests that the cognitive load imposed on an individual's information processing system in comprehending information is a function of both the inherent complexity of the information and the way in which it is presented (Paas, Renkl and Sweller, 2003; Sweller and Chandler, 1991). In CLT, cognitive load is conceptualised in three categories: intrinsic, extraneous and germane (Sweller and

Chandler 1991). Intrinsic cognitive load is the cognitive load attributable to the inherent complexity of the information presented. Comprehending a task analysis of for tying a shoelace would be considered to impose a lower cognitive load than comprehending a task analysis of flying an aircraft. Germane cognitive load is that required to process presentational features that facilitate comprehension (such as highlighting key terms). Extraneous cognitive load is that generated by processing presentational features that detract from comprehension. Sweller and Chandler (1991) identify two attributes of textual and graphical representations that can induce extraneous cognitive load. The first is where materials contain mutually-referring, disparate sources of information, such as text separated from a graphic or two separate pieces of text, have to be integrated resulting in what they term the split-attention effect. Mayer and Moreno (2003) describe the cognitive overload problem associated with this effect as representational holding; one of the representations has to be held in working memory whilst the second is processed and integrated with the first. The second attribute of written materials that causes extraneous cognitive load is where information is repeated. If this repeated information is integrated physically with essential information it results in what they term the redundancy effect, as the reader has no choice but to process it (Sweller and Chandler 1991).

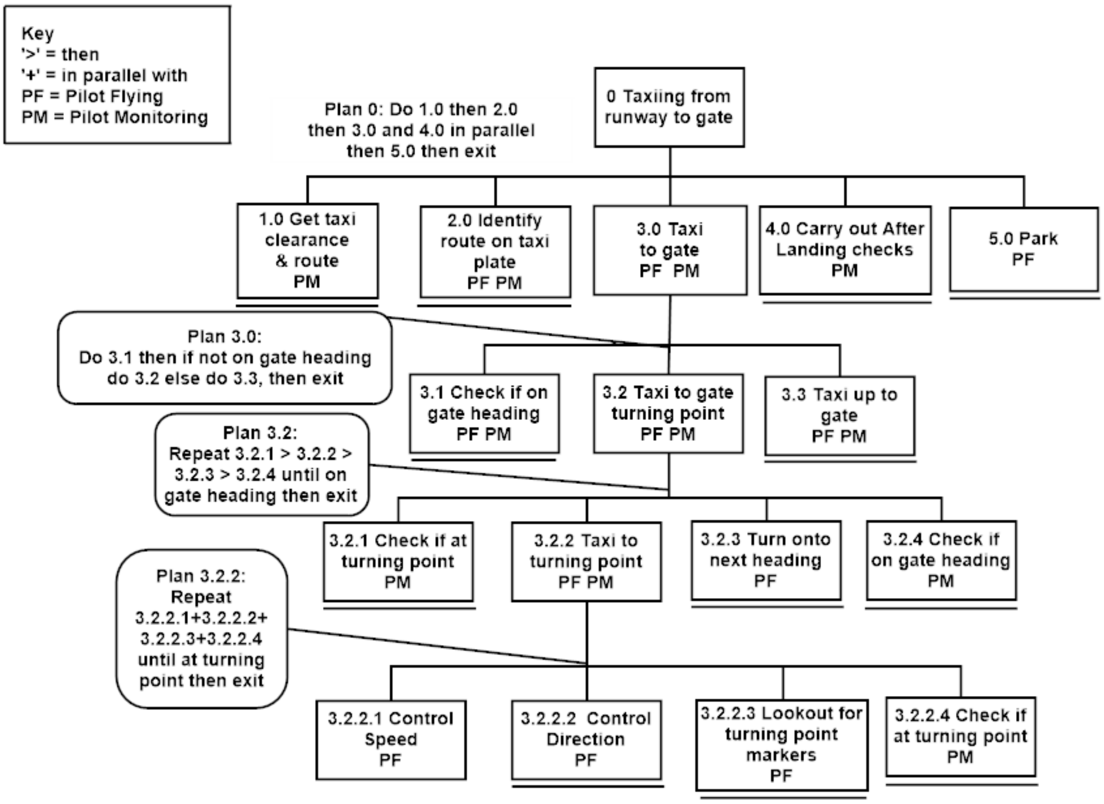
In this paper the constructs of representational holding and redundancy will be applied to the analysis of HTA notational constructs to determine if they impose extraneous cognitive load which could impact on the usability of HTA task representations, and to inform the revision of the notation to address any issues that are identified.

Analysis of HTA Task Representation Notations

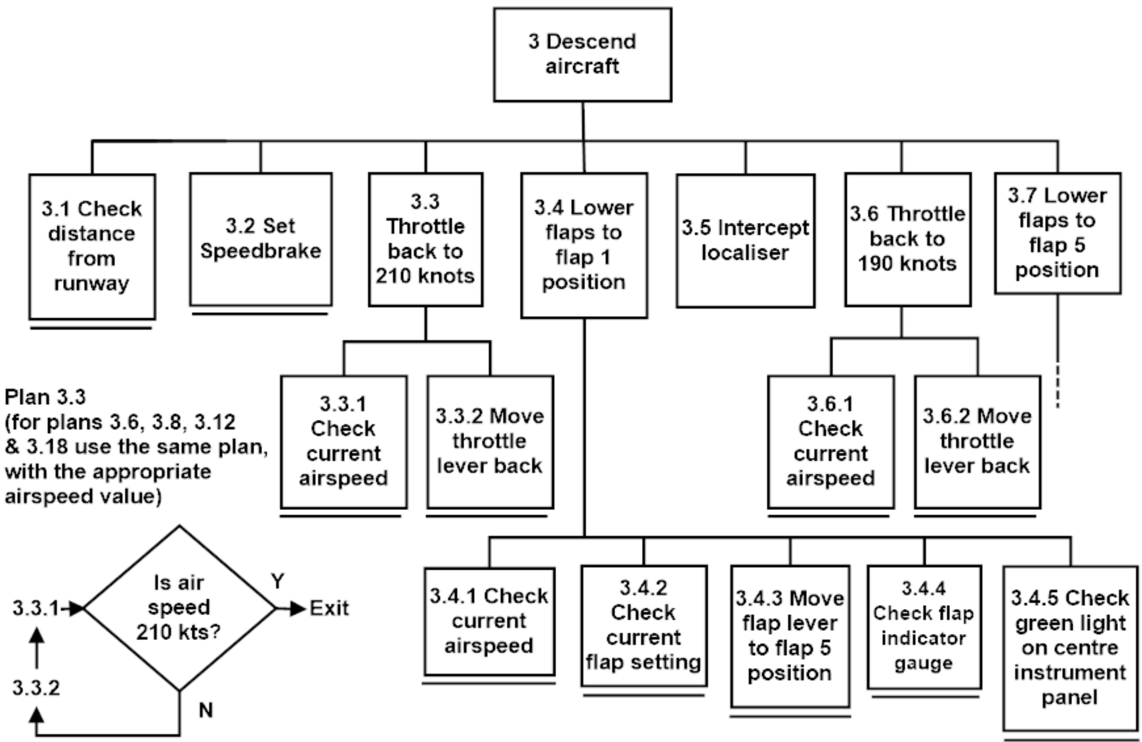
In this section, conventional graphical and tabular HTA notations are evaluated. Graphical representations are considered first, followed by tabular representations.

Figure 1 shows two examples of published HTA graphical representations. Figure 1a shows the top five levels of analysis for an aircraft taxying task adapted from Huddlestone et al (2014). This was produced during an investigation of aircrew tasks during current, two-crew flight operations. Figure 1b shows part of the analysis for the task of descending a Boeing 737 conducted by Marshall et al (2003). This was developed for use as a data source during the development of an error prediction methodology. The two components of the representations to be analysed are the plans and the sub-goal hierarchies.

The plan representations in Figure 1a have been adapted to illustrate alternative forms for text-based plan representations. The first plan in Figure 1a (Plan 0), is written in free text above the line of the decomposition that it applies to. Its close placement minimises the split attention problem in so far as the reader does not have to transfer their gaze far from the plan to the graphical structure that it applies to, but representational holding is required. The reader has to read and understand the plan and then hold it in working memory as they then apply it to the graphical structure to, in this case, understand which processes are carried out in sequence and which two are conducted in parallel within the overall sequence. The remaining plans in Figure 1a are written in boxes linked with lines to the node to which they apply. This is a useful approach where space precludes writing the plan above the horizontal line linking the sub-goals to which it applies. Representational holding is still required to integrate these plans with the sub-goals. Plans 3.2, 3.2.2 and 3.2.2.1 are written using symbolic shorthand, with “>” meaning “then” and “+” meaning in “parallel with”. This is a useful device where plans are long, particularly as the decimal code numbers for the



a. Partial HTA for an aircraft taxying task adapted from Huddlestone et al (2014)



b. Partial HTA for descending a Boeing 737, adapted from Marshall et al (2003).

Figure 1 Examples of graphical representations of a hierarchical task decompositions using conventional HTA notation.

sub-goals lengthen as the hierarchy deepens. However, the use of this shorthand can impose further extraneous cognitive load as the symbols have to be translated as part of the interpretation process. This may be minimal for experienced practitioners who are producing or reading HTAs frequently, but those less familiar with the notation (such as irregular users, novices or subject matter experts reviewing HTAs for accuracy) reference to a key may be necessary, adding further representational holding requirements. In figure 1b, a flowchart representation of a plan is used for sub-goal 3.3. This form of plan notation is also commonly used as an alternative to purely narrative plans. In the original articulation of the HTA method Annett et al (1971) refer to them as decision trees, although they were not included on the hierarchical representations. In this case, sub-goals 3.3.1 and 3.3.2 need to be memorised and then inserted into the flowchart to comprehend the meaning of the plan, which is another instance of representational holding being required. Finally, each of the plans contains the word "exit" to indicate that, once the sequence of sub-goals that the plan describes is complete, the next superordinate sub-goal is actioned. It could be argued that these "exit" statements are redundant, since this could be regarded as implicit in the nature of the hierarchical decomposition..

Considering the sub-goal hierarchies, it could be argued that one of the fundamental strengths of HTA and this form of representation is the elegant way in which abstraction is used to understand the structure of a complex task. At the first level of decomposition in Figure 1a, the taxiing task is represented as having a sequence of five sub-goals, each of which is simple to understand. At each of the lower levels of re-description of sub-goal 3 there are similarly simple to understand sequences of sub-goals. Interpreting the sub-goal hierarchy imposes cognitive load but this can be considered to be germane cognitive load, as it aids the understanding of a complex task.

Figure 1 also illustrates the conventions for indicating if sub-goals are further re-described. There are three possibilities with regard to re-description of a sub-goal: it is re-described in the same diagram, with subordinate sub-goals shown linked below it; it is re-described in another diagram, so it has links to sub-ordinate sub-goals but they are not showing in the diagram; or it is not re-described further and so has no sub-ordinate sub-goals to be shown. The notational convention commonly used in HTA diagrams is that if a sub-goal is not further re-described it is underlined, whereas if it is re-described on another diagram it does not have an underline. Reviewing Figure 1a, it can be seen that twelve underlines are required in order to show that only one sub-goal (3.2.2.1 Control speed, in the bottom left-hand corner of the diagram) is re-described elsewhere. Similarly, eleven underlines are required to show that only one sub-goal is re-described elsewhere in Figure 1b. It is also interesting to note that, in the full version of Figure 1b in Marshall et al (2003), 47 sub-goals are represented across two landscape pages. None of them are further re-described elsewhere but it takes 47 underlines to indicate this. Therefore, the underline convention results in extraneous cognitive load due to redundancy. When describing this notational convention in his book on HTA, Shepherd (2001, 91) states 'I have not maintained this convention in the rest of the book in order to enhance the clarity of the diagrams'. His point about clarity could be construed as a critique of the usability of the underlining nomenclature.

Also of interest in Figure 1b, is the fact that there are multiple instances of the same sub-goals having to be carried out; throttling back appears as sub-goals 3.3 and 3.6, whilst lowering flaps appears as sub-goals 3.4 and 3.7. In the full version of the HTA in Marshall et al (2003), there are a total of five instances of each of these sub-goals. As plan 3.3 suggests, the sequence of actions for throttling back is the same on each occasion, just the airspeed value is different on each occasion. Similarly, the actions for lowering flap are the same, just the required flap setting changes. Complete re-description of the throttling back and flap setting sub-goals for each instance of their use requires a total of 35 sub-goal boxes in the full HTA, whereas only seven of these are conveying new information. This can be considered as introducing ineffective cognitive load due to the redundancy effect. In addition, it introduces the possibilities of errors being made during the composition of the HTA, if the sequences are not copied correctly. This is a particular risk if modifications are made to such recurrent sub-goals, as the changes may not be copied into all instances of the recurrent structures.

Table 1 shows the top two levels of decomposition of the taxing task from Figure 1a represented in the tabular form proposed by Shepherd (1976). The sub-goals and plans can be copied into the table from the graphical HTA as they are expressed in exactly the same format. Table shows the re-description of sub-goal 3 below the re-description of the top level goal 0. The fact that a re-description of it will be found lower in the table is indicated by a ‘yes’ in the re-description column. Addition information and analysis of each sub-goal can be recorded in subsequent columns in the table. Table 1 shows just one column for notes, but the format can be adapted to support as many columns as are required to support the type of analysis being conducted. Notwithstanding the utility and simplicity of construction of the tabular format, it does have some issues in terms of extraneous cognitive load. Interpretation of the plans still requires representational holding as discussed above, and the exit statements are still redundant. In addition, the ‘Nos’ in the re-description column are analogous to the redundant underlines in the graphical format. Also, if recurrent operations were re-described as in the graphical format, the redundancy that this imposes would persist in the tabular form.

Table 1 Example of a tabular task analysis format as proposed by Shepherd (1976)

Conventional notation		**Further Redescribed**	**Notes**
0 Taxi from runway to gate Plan 0: Do 1 then 2 then 3 and 4 in parallel then 5 then exit.			
	1.0 Get taxi clearance route	No	
	2.0 Identify route on taxi plate	No	
	3.0 Follow taxi route to the gate	Yes	
	4.0 Carry out After Landing checks	No	
	5.0 Park at gate	No	
3.0 Follow taxi route to the gate Plan 3.0: Do 3.1 then 3.2 then 3.3 then exit.			
	3.1 Check if in gate heading	No	
	3.2 Taxi to gate turning point	Yes	
	3.3 Taxi up to gate	No	

In summary, this analysis has shown that the representation of plans in the conventional manner imposes extraneous cognitive load due to the requirement for representational holding. The use of exit statements in plans, the use of underlines to indicate when re-description stops, and the repetitive representation of recurrent operations all impose extraneous cognitive load due to redundancy.

Revised graphical and textual HTA notations

This section describes the revised graphical and textual HTA notations developed to address the extraneous cognitive load issues with the extant notations identified in the previous section, and provides the rationale behind their design. Three constraints were applied to the design of the notation. Firstly, the graphical and textual notations should be consistent; otherwise an extraneous cognitive load issue would be introduced due to the need to map one notation to the other when using both forms. Secondly, it should be possible to produce the representations using commonly available drawing and word processing packages (MS PowerPoint™ and MS Word™ were used as references). Thirdly, the representation of operations as sub-goals should be independent of the representation of their triggering and exit conditions. To illustrate the significance of this requirement, consider the task of hammering in a nail until it is flush. A description of this task might start with the nail first being held in place and struck once to locate in in the wood. Then, after its alignment is correctly, it is struck repeatedly until it is flush with the surface. The operation of striking the nail is the same in both cases. What differs, are its triggering and exit conditions, or to put it another way, the relationship between the subordinate operation (striking the nail) and the superordinate operation (hammering in the nail until flush).

A number of methodologies in the human factors and software engineering domains use the same graphical notation for representing hierarchies as that of HTA, but use additional graphical symbols to indicate different types of sequencing constructs, rather than written plans. Jackson Structured Programming (JSP) is a methodology developed for designing data processing software. Jackson (1974) states that it is based upon the principles of structured programming (Dijkstra 1968), the central tenants of which are that problems should be decomposed into hierarchical parts and that each level of decomposition should only contain sequence, repetition and selection constructs (analogous to HTA liner, cyclical and branching/selection plan constructs). Jackson's structure chart notation is used to portray hierarchical data structures and program structures, using these constructs. Structured Systems Analysis and Design Methodology (SSADM) Methodology was developed for the UK Government in the 1980s to support information systems design. It uses Jackson structure chart notation for a variety of modelling tasks. Notably, it uses an extended version of the notation to describe parallel events (Weaver et al 2003). Jackson Structured Design – Human Computer Interaction (JSD-HCI) was developed by Sutcliffe and Wang (1990) to facilitate the integration of task analysis and human computer interface design into Jackson System Development (Jackson 1983). It uses Jackson structure charts to represent task descriptions and function allocations. In a similar vein, the Method for Usability Engineering (MUSE) was developed by Lim and Long (1994) to provide a structured approach to the integration of human factors activity into the systems development lifecycle. They use Jackson structure chart notation for a wide variety of modelling tasks, including task modelling, and have extended the notation to cater for concurrent and non-linear sequences and to describe the conditions under which cyclical tasks are executed. Conventional HTA plan constructs and the corresponding Jackson structure chart-based notations used by these four methods are shown in Figure 2 (columns 1 and 2),

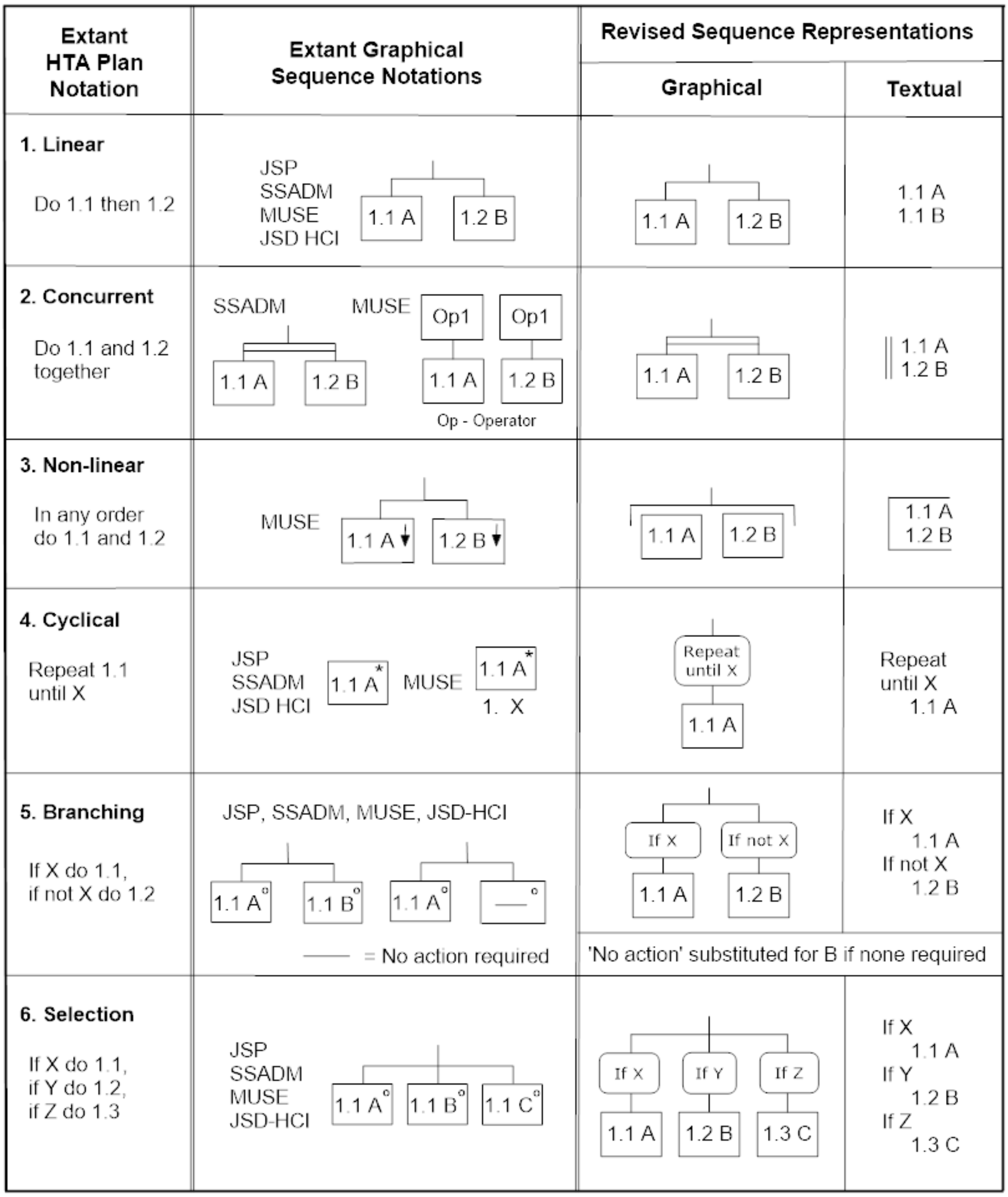


Figure 2 Extant HTA plan notation, extant graphical sequence notation and new HTA sequence notations for the six conventional HTA plan types.

In all of the methods reviewed, the notation for linear sequences formed the baseline notation from which all the others were developed by the addition of other symbology. This approach was adopted for the development of the new HTA symbology. Linear sequences are shown using the conventional HTA representation for decomposition, with the textual equivalent being a simple list. (Figure 2, row 1). The MUSE notation for concurrent operations requires the production of separate charts for each operation, so was rejected. However, SSADM addresses this in a simple way by using a replacing the single horizontal line connecting operations with a parallel line. As this could easily be replicated in the textual representation by modifying the formatting of the cell border in a table, this symbology was adopted (Figure 2, row 3). Only MUSE had a graphical representation for non-liner sequences, with a downward pointing arrow in the top right-hand corner of the box for each operation in the sequence. However, this representation conflates the representation of sub-goals with the representation of triggering and exit conditions, so was rejected in all cases. In the case of non-linear sequences, a square bracket to surround the operations was used instead as it groups the sub-goals whose order of execution can be swapped, yet distinguishes the grouping from a linear or parallel sequence. Also it could easily be replicated in the textual notation by the use of emboldened in table cell border formatting. MUSE was also unique in that it was the only method in which there was a representation of triggering conditions for cyclical representations, listing them under the operation box (Figure 2, row 4). Since triggering conditions determine if a sub-operation is executed, and as such define the link between a sub operation and its super-ordinate operation, it was considered more logical to place the condition statement (in the form 'repeat until X') above the sub-goal to which they applied (Figure 2, row 4). A rounded box was used to distinguish condition statements from sub-goals. In the textual format, the condition statement is written above the sub-goal. The same formats were used for representing branching and selection constructs, with the triggering condition statements written in the form 'if condition' (Figure 3, rows 5 and 6). Branching structures commonly represent two alternative courses of action, such as shaking martini if the customer is James Bond, or stirring it if the customer is not James Bond. However there are situations where no alternative course of action is required. For example, the alternative to turning on car windscreen wipers if it is raining is to take no action if it is not raining. In all of the alternative notations examined 'no action' is represented by a horizontal line in operation box. As this constitutes just another piece of notation to be learn, simply writing 'no action' was favoured. It was noted that none of the representations in the other methods examined represented conditions for selection and branching constructs, which would suggest that task models produced using these notations would be incomplete.

The redundancy effect caused by repeated re-description of sub-goal hierarchies for operations that recur in the task hierarchy (referred to as recurrent operations in this paper) can be solved very simply. The operation is re-described in a sub-goal hierarchy separate from the main sub-goal hierarchy, and then referenced wherever it is required in the main sub-goal hierarchy. This is conceptually equivalent to the approach taken in the Goals, Operators, Methods and Selection Rules (GOMS; Card, Moran and Newall 1983) methodology whereby high-level methods (the equivalent of operations) reference lower-level methods. It is also analogous to the sub-routine construct used in software languages which allows for sections of program code to be written once and then called from many different points in the main program. The graphical notation used for defining and referencing recurrent operations is shown in Figure 3. A box with parallel lines on each side was selected to represent recurrent operations as it is the symbol used to represent sub-routines in flowchart notation and is commonly available as a standard symbol in graphics packages such as Microsoft PowerPoint™. Also, the same notation can be easily constructed in the textual form using line formatting for cells in a table. Recurrent operations are uniquely numbered in the form Rn, where n is an integer number stating from 1, so if there were five recurrent operation identified in a task analysis they would be numbered R1-R5. This avoids any potential ambiguity of sub-goal numbering across multiple hierarchies. The final aspect of the notation is the use of parameters. Recurrent operations will often contain cyclical or branching/selection structures. However the actual values to be used in the conditions statements will typically be different for each situation that they are referenced. For example, in analysing the task of driving a car, a recurrent operation ‘accelerate to speed limit’ might be identified. When describing driving through a village this operation would be referenced as ‘accelerate to 30 mph’, whereas the description of driving on a motorway would reference it as ‘accelerate to 70 mph’. The term ‘speed limit’ serves as a placeholder in the condition descriptions of any cyclical and branching/selection structures in the definition, for which actual values which are substituted when the recurrent operation is referenced. Such placeholders are referred to in the definition of recurrent operations as ‘parameter names’, with the values that replace them in recurrent operation references being termed ‘parameter values’.

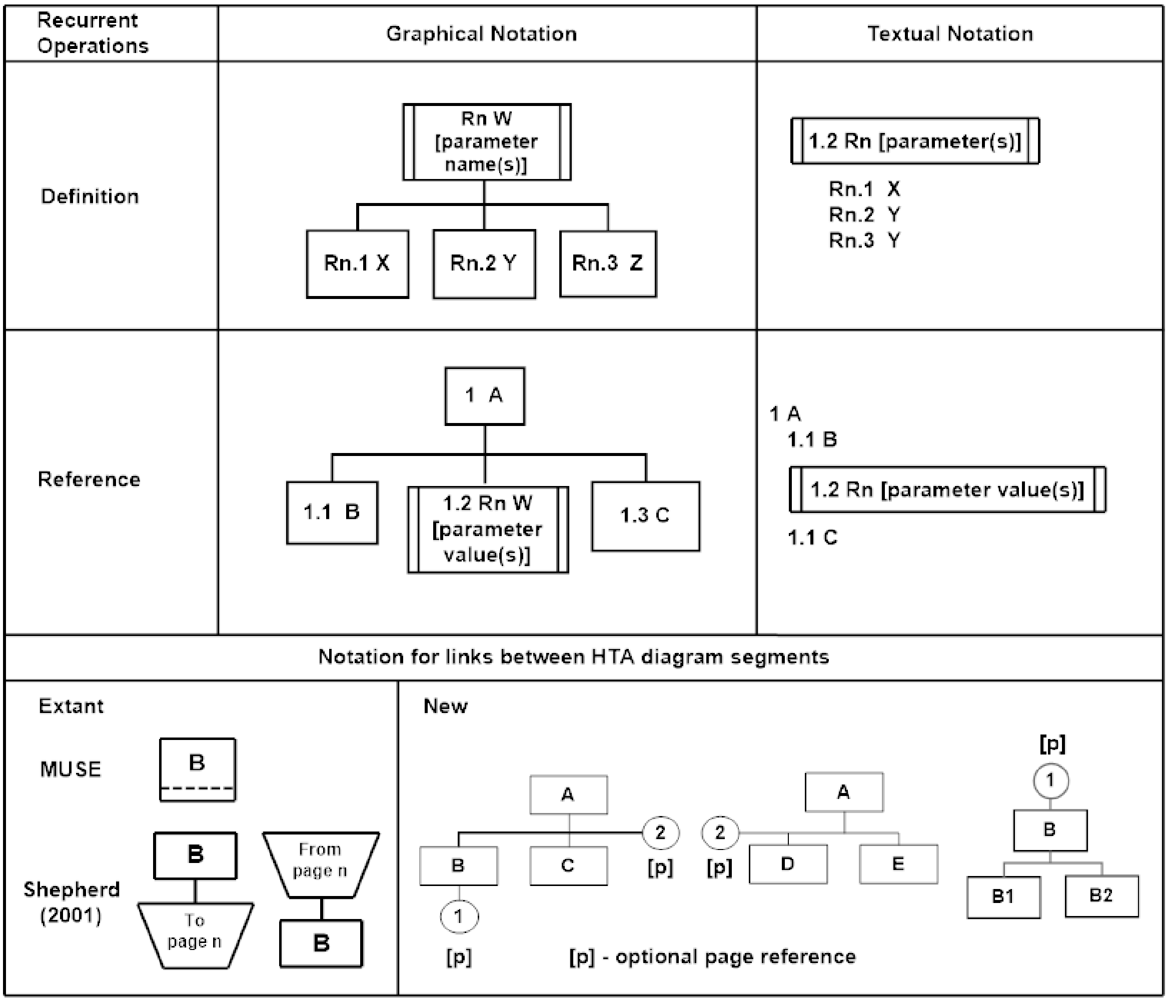


Figure 3 Notational representations for recurrent operations and links between HTA diagram segments

The issue of the redundancy of underlines used to indicate that a sub-goal is not re-described any further in the graphical representations has been partially addressed by Lim and Long (1994) in the MUSE notation by using a horizontal dotted line at the bottom inside (Figure 3). Whilst recommending the use of underlines for the cessation of re-description, Shepherd (2001) also suggests the use of lozenge shaped page connectors to show links across pages (Figure 3). The notation selected extends

Shepherd's suggestion by using numbered circles with an optional page number if required. This caters for the possibility of more than one sub-goal being further re-described on the same page. For consistency, the same link notation is applied to lateral splits in the graphical representation. In the textual format a bold '**R**' is written at the end of a sub-goal that is re-described further, as used by Annett et al (1971) in their original exposition of the method.

Examples of the use of the revised notations

Figure 4 shows an HTA for the 'taxi from runway to gate task' presented in the revised graphical notation. The decomposition of the 'control speed task' has been included at the bottom to illustrate the use of the selection construct. There are a number of points worth noting about this representation, compared with the version in conventional HTA notation presented at Figure 1a. The removal of plan statements and underlines reduces the number of graphical elements in the diagram and increases the amount of white space, which increases visual clarity. It is also easy to see the sequencing constructs. At the first level of decomposition, the parallel connection between the third and fourth sub-goals stand out, as does the parallel connection between the sub-goals in the lowest level of the upper hierarchy. Also, the cyclical sequences become very clear, due to the white space that results from the conditions statements in the rounded box being placed between the sub-goals and the superordinate goal which they re-describe. They literally link the sub-goals and the superordinate goal. It is also easy to see the one off-page link, indicating that the bottom left sub-goal is re-described elsewhere. The re-description of the '3.2.2.1 Control speed 'sub-goal shows the use of the selection construct (Figure 4). The selection conditions for each of the sub-goals in the selection construct can be read in the rounded boxes immediately above each sub-goal.

In conventional graphical HTA representations the sequencing of sub-goals can only be determined by reading the plans and then applying them to the sub-goals they refer to, where as in the new notation they can be read directly from the hierarchical links in the diagram. By contrast, in HTA representations using the new graphical notation, the sequencing of any one sub-goal can be determined simply by inspecting the representation of the link between it and its superordinate sub-goal/goal; there is no need to look elsewhere for information. One could argue that the sequencing constructs have been promoted visually to equal status with the sub-goals in the graphical representation.

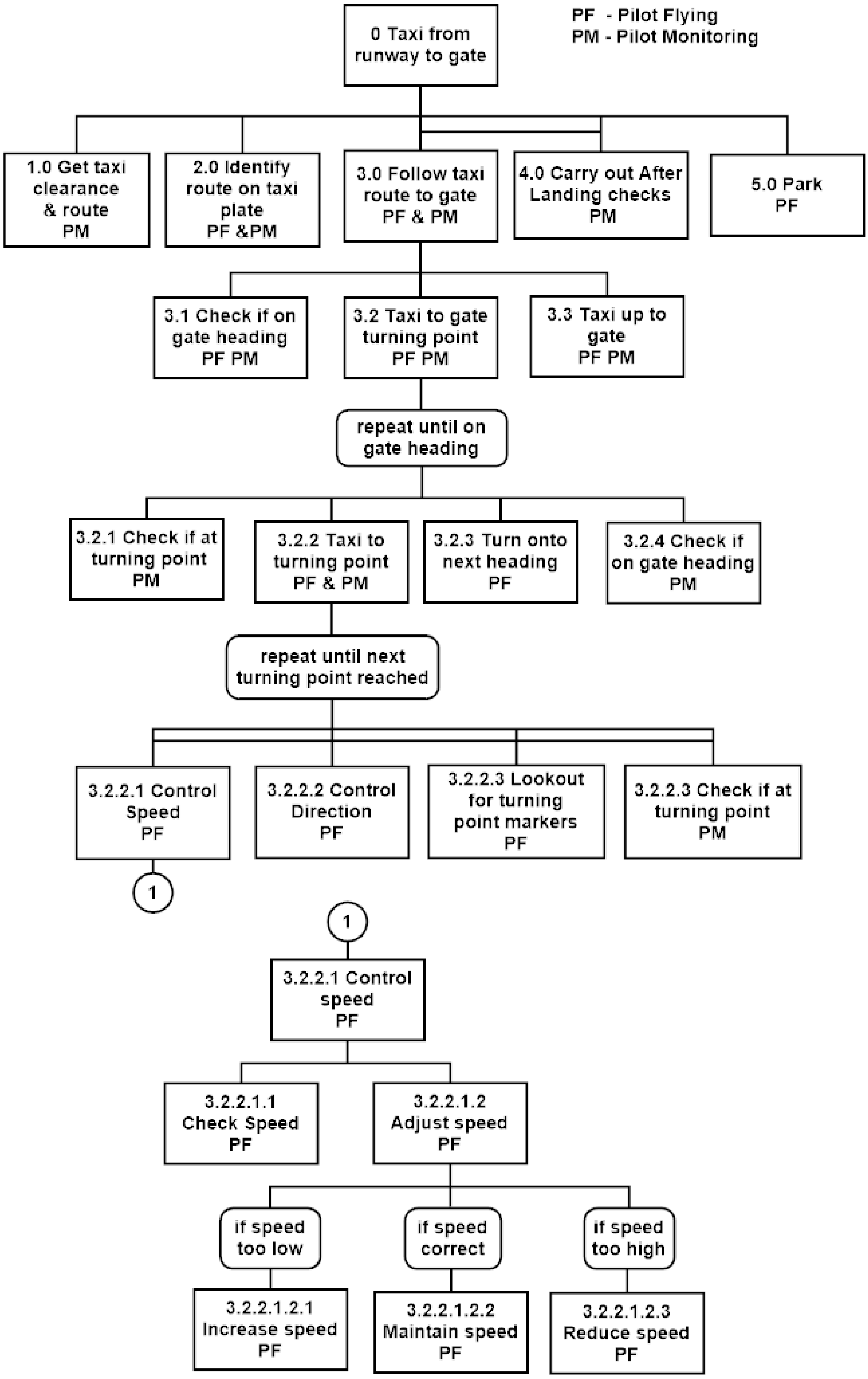


Figure 4 HTA for the 'Taxy from runway to gate' task in the revised graphical notation

Figure 5 shows the 'Descend aircraft task', originally presented in Figure 1b, recast in the new graphical notation, illustrating the use of the notation for recurrent sub-goals. In the top part of Figure 5, multiple instantiations of the recurrent sub-goal

‘R1 Throttle back’ as sub-goals 3.3 and 3.6 in the main hierarchy can be seen, with the required speed showing as a parameter. A second recurrent sub-goal (R2) is shown for the ‘lower flaps’ sub-goal. The re-description of R1 is shown in the lower part of Figure 5. It has the parameter ‘required speed’ which is shown by name in the top level box in the hierarchy and is used in the condition controlling the iteration of the lowest two sub-goals. The use of the recurrent sub-goal notation results in significant simplification of the sub-goal hierarchy compared with the version in conventional notation shown in Figure 1b. Given that in the full version of the HTA in Marshall et al 2003 there are five occurrences of the ‘throttle back’ and ‘lower flaps’ tasks, the use of the recurrent sub-goal notation would substantially reduce the complexity of the HTA representation.

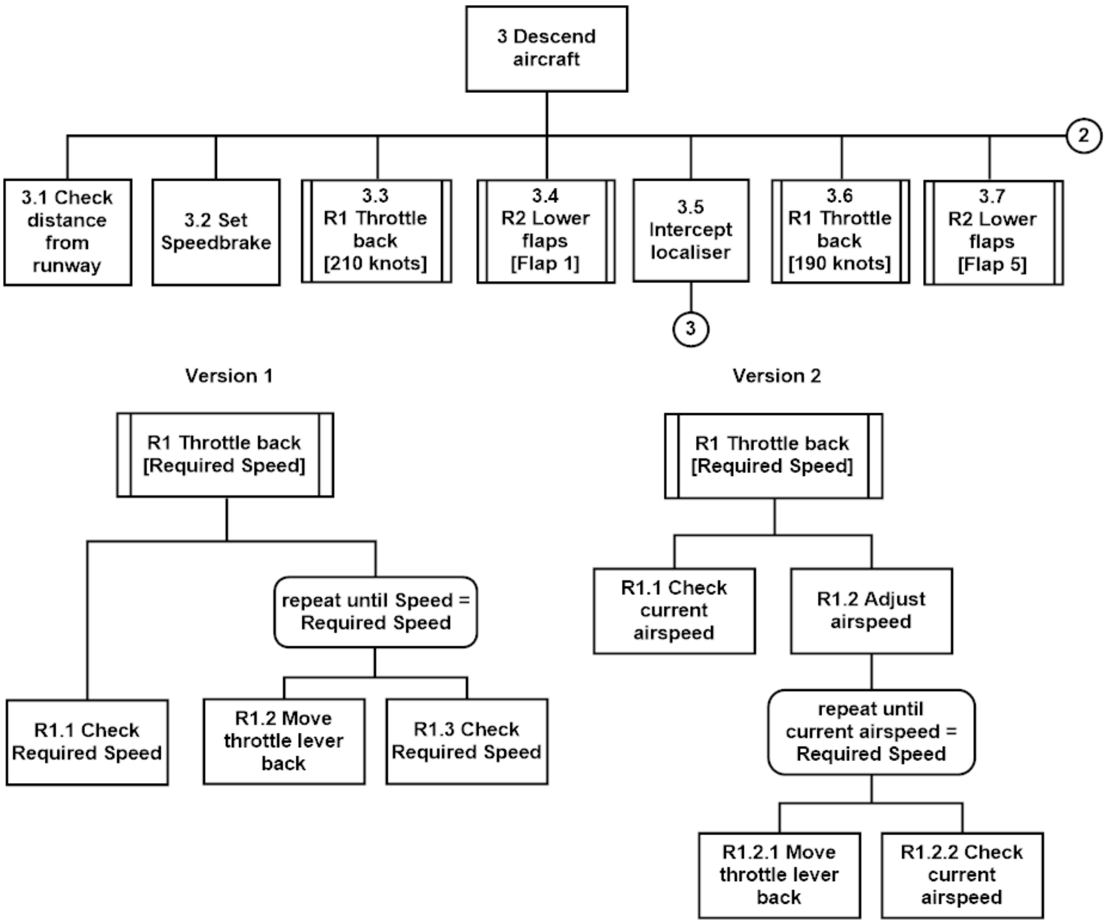


Figure 5 Illustration of the use of the notation for defining and instantiating recurrent sub-goals.

Two versions of the definition of the ‘R1 Throttle back’ definition sub-goal hierarchy are shown in Figure 5. They are in fact logically equivalent, the only difference being that an additional sub-goal (R1.2 Adjust speed) has been introduced which describes the cyclical sequence below it. It is left to the reader to determine which is the most aesthetically pleasing and which is the simplest to read. From a CLT perspective the question is whether the introduction of the additional sub-goal into the structure constitutes extraneous or germane cognitive load. The key point here is that the cognitive load of the representation is not determined purely by the form of the notation, but how the notation is used in representing the task. This is a known reliability issue that has been documented in the literature (Stanton 2013).

Table 2 shows the use of the revised textual notation in the tabular representation of the HTA for the elements of the ‘taxi from runway to gate’ task. It has been modified from the original version to illustrate the use of the recurrent sub-goal notation (sub-goal 3.2.3) and the non-linear sequence construct (sub-goal 4.0). As with the graphical HTA notation, it is easy to spot the different types of sequencing constructs.

Table 2 Tabular representation of elements of the ‘Taxy from runway to gate’ task, presented in the revised notation.

Key	‖ Do in parallel	[Do in any order	‖ Recurrent operation ‖
Sub-goal Hierarchy **(R)** indicates sub-goals that are re-described further down in the table			**Notes**
0 Taxi from runway to gate			
‖	1.0 Get taxi clearance route		
‖	2.0 Identify route on taxi plate		
‖	3.0 Follow taxi route to the gate **(R)**		
‖	4.0 Carry out After Landing checks**(R)**		
‖	5.0 Park at gate		
3.2 Taxi to gate turning point Repeat until on gate heading			
	3.2.1 Check distance to turning point		
	3.2.2 Taxi to turning point **(R)**		
	‖ 3.2.3 R1 Turn onto next heading ‖		
	3.2.4 Check if on gate heading		
3.2.2.1.2 Adjust speed			
	If speed too low 3.2.2.1.2.1 Increase speed		
	If speed correct 3.2.2.1.2.2 Maintain speed		
	If speed too high 3.2.2.1.2.1 Decrease speed		
4.0 Carry out after landing checks			
[	4.1 Turn off weather radar		
[	4.2 Raise flaps		

One way to get a sense of the difference between the conventional and the new textual notations in the tabular form is to imagine having to give a set of instructions on how to achieve one of the higher level sub-goals to someone sat across the table who cannot see the text. Taking the first Goal 0 as an example in Table 2, the text from the new notation could simply be read out: “Get [the] taxi clearance route, identify [the] route on [the] taxi plate, then in parallel, follow the taxi route to the gate and carry out After Landing checks, then park at the gate.” The same could not be said for the conventional notation version in Table 1, where each plan has to be interpreted and applied to the list of sub-goals first, in some cases after decoding the abbreviated notation. It is contended that this difference reflects a useful reduction in intrinsic cognitive load in reading the new notation which should facilitate both the development and interpretation of tabular HTAs.

Discussion

The use of cognitive load theory has facilitated the identification of attributes of conventional HTA graphical and textual notations which impose extraneous cognitive load due to representational holding and redundancy when reading task decomposition hierarchies. It has also proved useful in guiding the refinement of the notational constructs to address these issues. However, the key issue is whether the revisions to the notations enhance their usability in terms of efficiency, effectiveness and satisfaction from the perspective of supporting communication and the development of task models.

The use of the notation for recurrent operations should result in the production of less complex models which have lower therefore have lower intrinsic cognitive load, whilst also reducing the extraneous cognitive load associated with redundancy. Similarly, the graphical visualisation of sequencing and triggering condition constructs proximal to the sub-goals that they relate to should eliminate much of the extraneous cognitive load related to representational holding. If these assertions hold true then readers of the task hierarchies should have more cognitive resources available to focus on their primary task which was the purpose for reading the hierarchies in the first place. This in turn should lead to more effective and efficient conduct of their tasks. This would go some way to addressing industry concerns about the difficulties faced by non-analysts in reading task analysis outputs which Stanton and Diaper (2004) report. Similarly, the reductions in complexity which the use of recurrent operation notation should afford should manifest itself in more expeditious production of task hierarchies. Reductions in the cognitive load associated with reading task hierarchies should also facilitate their utility for communication within teams during their production which should also impact positively on the efficiency and effectiveness of the production task. If both analysts and untrained stakeholders are able to carry out their tasks both more efficiently and effectively, one could posit that task would increase. However, for these benefits to be realised the differences in cognitive load, both intrinsic and extraneous would have to be of a sufficient magnitude to make a practical difference to the communication and development related tasks. All of that said, cognitive load of a task hierarchy is only in part attributable to the notation used. The intrinsic cognitive load is related to complexity of the representation which is a function of both the complexity of the task being represented and the way in which the analyst uses the notation to represent the task. The dearth of style guidelines for developing ‘good’ task hierarchies is concerning.

Conclusions

The aim of this paper was to take a step towards improving the usability of a task analysis method. Analysis of the conventional graphical and tabular HTA notations using CLT constructs has identified that some of the notational constructs impose extraneous cognitive load when reading HTA task representations. Revised graphical and textual notations have been developed to address these issues, and from a theoretical perspective, it has been shown that they should impose less extraneous cognitive load when reading HTA task representations produced using these notations. Therefore, it is contended that the development of these notations constitutes a potential step forwards in the usability of HTA notation.

However, some key questions have yet to be answered. Empirical evidence needs to be gathered to determine if the use of these notations results in significant improvements in the readability of task representations which manifest in shortened time to both construct and interpret HTA representations produced using the new notation compared with equivalent representations produced using conventional notation. Studies are also required to determine if the use of these notations enhances the learnability of HTA as a method. More ambitiously, it would be interesting to investigate if the use of these notations yields improvements in the effectiveness and efficiency of HTA use in professional settings, both academic and commercial.

More broadly, if the application of CLT in this study can be likened to putting an arrow in the quiver of techniques used to evaluate and enhance the usability of ergonomics methods, it is hoped that this work inspires colleagues in the domain to place further arrows in the quiver and to fire them at a wider range of methodological targets than simply representational forms. The development of a generic usability framework for ergonomics methods might be one such arrow. The development of style guidelines for the use of graphic notation within methods might be another.

References

Adams, A.E., Rogers, W.A. and Fisk D.F. 2013 “Skill Components of Task Analysis.” Instructional Science (2013) 41: 1009-1046

Ainsworth, L. and Marshall, E. 1998 “Issues of practicability in task analysis: preliminary results from two surveys.” Ergonomics (41) 11 : 1607-1617

Annett, J. 2004 “Hierarchical Task Analysis” in The Handbook of Task Analysis for Human-Computer Interaction. edited by Neville Stanton and Dan Diaper, 67-82. Mahwah, NJ: Laurence Earlbaum Associates

Annett, J., Duncan, K.D., Stammers, R.B., Gray, M.J. 1971. Task Analysis. Department of Employment Training Information Paper 6. HMSO, London.

Balbo 2004, S., Ozkan, N. and Paris. 2004 “Choosing the Right Task-modelling Notation: A Taxonomy” in The Handbook of Task Analysis for Human-Computer Interaction. edited by Neville Stanton and Dan Diaper, 445-466. Mahwah, NJ: Laurence Earlbaum Associates

Card, S.K. Moran, T. and Newall, A. 1983 The psychology of human computer interaction. Hillsdale NJ: Laurence Earlbaum Associates.

Dijkstra, E. W. 1968. "Letters to the editor: go to statement considered harmful". Communications of the ACM 11 (3): 147–148.

ISO 2015 Ergonomics of human system interaction – Part 11 Usability: Definitions and concepts ISO/DIS 9241-11 Geneva: ISO

Jackson, M. 1975. Principles of Program Design, London : Academic Press

Jackson, M. 1983 System Development, Englewood Cliffs, NJ: Prentice Hall

Lim, K.Y. and Long, J. 1994 The MUSE Method for Usability Engineering. Cambridge: Cambridge University Press

Marshall, A., Stanton, N., Young, M., Salmon, P., Harris, D., Demagalski, J.,Waldmann, T. and Dekker, S. 2003. Development of the Human Error Template – a new methodology for assessing design induced errors on aircraft flight decks. Final Report of the ERRORPRED Project E!1970 (August 2003), London: Department of Trade and Industry

Mayer, R.E. and R. Moreno. 2003. "Nine ways to reduce cognitive load in multimedia learning." Educational Psychologist. 38(1): 43-52

Miller, G.A. Gallanter, E. and Pibram, K. 1960 Plans and the Structure of Behaviour. New York: Holt

Ormerod, T.C., J. Richardson and A.Shepherd. 1998. "Enhancing the usability of a task analysis method: a notation and environment for requirements specification." Ergonomics. 41(11): 1642-1663

Paas,F., A. Renkl, and J. Sweller. 2003. "Cogntive load theory and instructional design: Recent developments." Educational Psychologist. 38(10): 1-4

Paris,C. Balbo S., and Ozkan, N. 2000 "Novel uses of task models: Two case studies" Cognitive Task Analysis edited by Schraagen, J.M.C. PitmanS.E. and Shalin. 261-274 Mahwah, New Jersey: Lawrence Erlbaum Associates

Patrick, J. Gregov, A. and Halliday, P. 2000 "Analysing and training task analysis." Instructional Science 28: 51-79

Shepherd, A. 1976. An improved tabular format for task analysis. Occup. Psychol. 49, 93–104

Shepherd, A. 2001 Hierarchical Task Analysis. New York: Taylor and Francis

Stanton, N. A. 2004. The Psychology of Task Analysis Today. In: D. Diaper and N. A. Stanton (Eds) The Handbook Of Task Analysis For Human-Computer Interaction. 569-584. Mahwah, New Jersey: Lawrence Erlbaum Associates.

Stanton, N.A. 2006: Hierarchical task analysis: Developments, applications, and extensions. Applied Ergonomics 37, 55–79

Stanton, N. and Daiper, D, eds. 2004 The Handbook of Task Analysis for Human-Computer Interaction. Mahwah, NJ: Laurence Earlbaum Associates.

Stanton, N.A., Salmon, P.M., Walker, G.H., Baber, C. and Jenkins, D. 2005. Human Factors Methods: A Practical Guide for Engineering and Design (first edition). Aldershot: Ashgate.

Stanton, N. A., Salmon, P. M., Rafferty, L. A., Walker, G. H., Baber, C. and Jenkins, D. 2013. Human Factors Methods: A Practical Guide for Engineering and Design (second edition). Ashgate: Aldershot.

Stanton, N. A., & Young, M. S. 1998. A guide to methodology in ergonomics: Designing for human use (first edition). London: Taylor & Francis.

Sutcliffe, A.G.and Wang, I991 "Integrating Human Computer Interaction with Jackson System Development." The Computer Journal 34 (2) : 132-142

Sweller, J. and P. Chandler. 1991" Evidence for cognitive load theory" Cognition and Instruction. (4): 351-362

van Welie, M. van der Veer, G. and Koster, A. 2000 "Integrated Representations for Task Modelling." Proceedings of the Tenth European Conference on Cognitive Ergonomics. Paris: EACE

Weaver, P., Lambrou, N. and Wakeley, M. 2002. Practical Systems Development using SSADM, London: Prentice Hall