Dual Pilot and Single Pilot Operations – Hierarchical Task Decomposition Analysis of Doing More with Less

John Huddlestone(✉), Don Harris, Dale Richards, Steve Scott, and Rod Sears

Coventry University, Coventry, UK
{john.huddlestone,don.harris,dale.richards,steve.scott}@coventry.ac.uk, wilts8874@hotmail.co.uk

Abstract. An operational concept for single pilot operations is being developed as part of the work of the Future Flight Deck Technologies project. The underpinning construct is that multiple single-pilot aircraft will be supported by a ground-based team using one or more ground stations to interact with the pilots and their aircraft. Concept development required detailed analysis of two-pilot operations to identify the functions and interactions that the second pilot engages in, to facilitate their reallocation or adaptation within the new system architecture. The application of Hierarchical Task Analysis (HTA) in this context revealed a number of issues related to the representation and interpretation of sequencing information in plans in HTAs presented in graphical form. This paper identifies these issues and presents a new graphical notation, derived from software design notations, for presenting hierarchical task decompositions. The use of the notation is illustrated with analysis examples from different phases of flight.

Keywords: Hierarchical task decomposition · Hierarchical task analysis · Subgoal hierarchy · Plans · Single-pilot operations

1 Introduction

As part of the Future Flight Deck Technologies project, a concept of operations is being developed for Single Pilot Operations (SPO). The high-level concept that is being explored is based upon the notion of multiple single-pilot aircraft being supported by a ground-based team (or teams) using one or more ground stations to interact with the pilots and their aircraft, along with technological enhancements to the flight deck architecture to simplify tasks. To develop such a concept, consideration has to be given to how the functions currently executed by the second pilot on the flight deck can be redistributed within the new system architecture. This in turn requires a detailed analysis of current, two-pilot operations to identify the functions that have to be executed, the allocation those functions within the system, and the interactions between various actors executing those functions.

Due to the scale and complexity of the flying task, Hierarchical Task Analysis (HTA) was selected as the first of a number of analytical techniques to be applied to

D. Harris (Ed.): EPCE 2015, LNAI 9174, pp. 365–376, 2015.
DOI: 10.1007/978-3-319-20373-7_35

develop a detailed understand of two-pilot operations. A number of highly experienced pilots were consulted as Subject Matter Experts (SMEs) to inform the analysis and to review the outputs. During the analysis process, a number of issues were identified in both the development and the interpretation of graphical HTAs, notably with the representation and interpretation of plans. In response to these issues, a purely graphical notation for representing hierarchical task decompositions was developed, which was derived from notations used within the software engineering domain for describing data structures and software designs.

In this paper we first describe the issues that were identified in the development and interpretation of graphical representations of HTAs. These are illustrated with examples of analyses from different phases of flight. We then explain the new graphical notation that was developed. Finally, the application of this notation is illustrated by recasting the examples used in the first section using the new notation.

2 Issues in the Representation of Hierarchical Task Decompositions Using Conventional HTA Graphical Notation

In the original paper that describes the approach for conducting HTA [1], Annett cites goal directed behaviour as the theoretical underpinning of the approach. The principles of analysis ate that tasks are conceptualised as operations which can be defined in terms of their goals, and operations can be broken down into sub-operations, each of which can be defined by a sub-goal. Furthermore, the relationship between operations and sub-operations is hierarchical. Annett also highlights that, whilst many tasks are proceduralised (i.e. their sub-goals have to be achieved in a fixed sequence), this is not always the case [1]. The implementation of these principles in HTA results in tasks being represented as sub-goal hierarchies in which sub-goals are linked to their super-ordinate sub-goal or goal by plans, which specify their triggering conditions and sequencing.

An example of the format of a sub-goal hierarchy represented in conventional HTA notation is shown in Fig. 1. This particular example is a simplified version of the top three levels of analysis conducted on the taxiing to the gate task in the FFD project. Taxi to the gate is show as the overall goal in the box at the top and is numbered 0. At the first level of decomposition, the sub-goals are numbered from 1 to *n* for cross referencing purposes (1-5 in this case). In subsequent levels the numbering is related to the sub-goal above. In this example sub-goal 3 is decomposed into three sub-goals, numbered 3.1-3.3. The underlines under sub-goals indicate that that there is no further decomposition, so in this example only sub-goal 3.2 will be analyzed further.

In this form, the sub-goal hierarchy is very clear and easy to read. However, it becomes more complex when plans are added. Conventionally, sequencing and trigger condition information is represented for the set of sub-goals below a superordinate goal in the form of a plan. Figure 2 shows an expanded version of Fig. 1 which has plans superimposed on it using a variety of notational alternatives.

Plan 0 in Fig. 2 shows the use of free text written just above the sequence of sub-goals which it refers to, and describes a sequence which has two sub-goals that are met

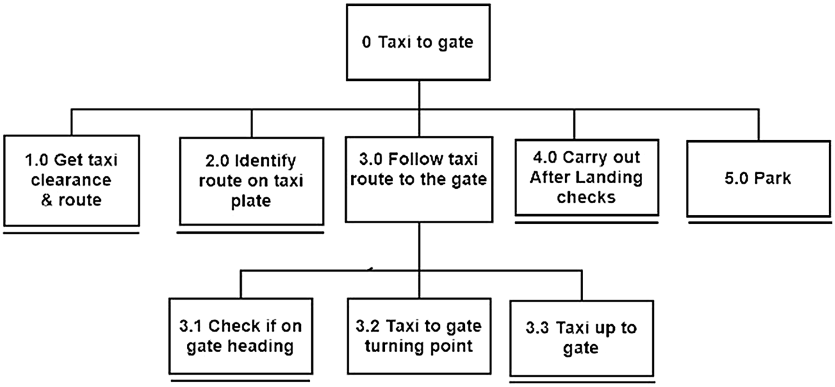

0 Taxi to gate
1.0 Get taxi clearance & route
2.0 Identify route on taxi plate
3.0 Follow taxi route to the gate
4.0 Carry out After Landing checks
5.0 Park
3.1 Check if on gate heading
3.2 Taxi to gate turning point
3.3 Taxi up to gate

Fig. 1. Example sub-goal hierarchy in conventional HTA notation

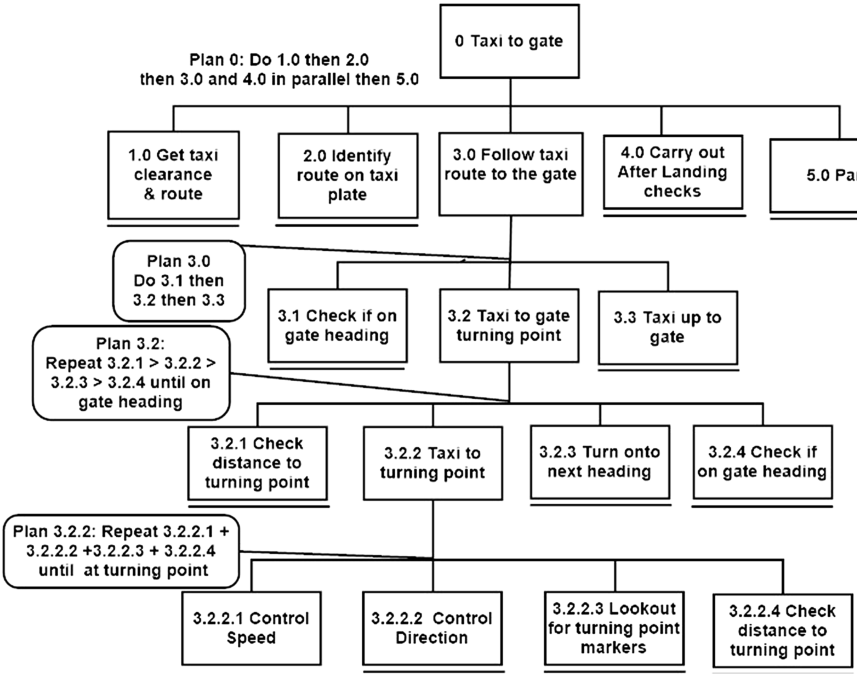

0 Taxi to gate
Plan 0: Do 1.0 then 2.0 then 3.0 and 4.0 in parallel then 5.0
1.0 Get taxi clearance & route
2.0 Identify route on taxi plate
3.0 Follow taxi route to the gate
4.0 Carry out After Landing checks
5.0 Park
Plan 3.0 Do 3.1 then 3.2 then 3.3
3.1 Check if on gate heading
3.2 Taxi to gate turning point
3.3 Taxi up to gate
Plan 3.2: Repeat 3.2.1 > 3.2.2 > 3.2.3 > 3.2.4 until on gate heading
3.2.1 Check distance to turning point
3.2.2 Taxi to turning point
3.2.3 Turn onto next heading
3.2.4 Check if on gate heading
Plan 3.2.2: Repeat 3.2.2.1 + 3.2.2.2 +3.2.2.3 + 3.2.2.4 until at turning point
3.2.2.1 Control Speed
3.2.2.2 Control Direction
3.2.2.3 Lookout for turning point markers
3.2.2.4 Check distance to turning point

Fig. 2. Example of a sub-goal hierarchy with plans attached.

in parallel (3.0 and 4.0). This approach works when there is enough space to write the plan in this form. Where there is insufficient space available, the format for Plan 3.0 can be used where uses the same free text format but is enclosed in a rounded box (to differentiate it from a sub-goal) with a link to the node above the sequence of sub-goals to which it applies. As the analysis deepens, even this approach becomes problematic, as the lengthening sub-goal numbers require even more space in the plans. To overcome this, it is common practice to use an abbreviated notation, as shown in plans 3.2 and 3.2.2. The “>” symbol in plan 3.2 is read as “then”, so the plan describes the repetition of a linear sequence of operations until the aircraft is on the gate heading. The “+” symbol in plan 3.2.2 is read as “in parallel with”, so the plan describes the repetition of a set of operations conducted in parallel until the aircraft is at a turning point.

The underlines in Fig. 2 indicate that only one sub-goal (3.2.2.1) is to be decomposed further. Figure 3 shows how this decomposition would be shown on a separate page. Sub-goal 3.2.2.1 is repeated and then decomposed. The final plan (Plan 3.2.2.1.2) describes what is referred to as a selection or exclusive plan, that is to say a set of optional sub-goals only one of which is executed, dependent on the associated condition. In this case the speed of the aircraft is either increased, maintained or decreased, dependent on whether the current speed of the aircraft is too slow, correct or too fast.

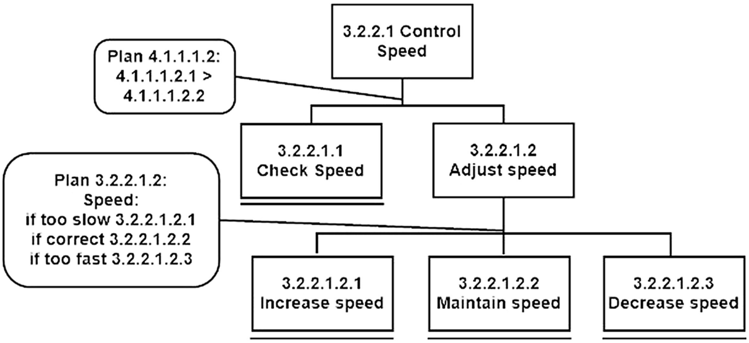

3.2.2.1 Control Speed
Plan 4.1.1.1.2: 4.1.1.1.2.1 > 4.1.1.1.2.2
3.2.2.1.1 Check Speed
3.2.2.1.2 Adjust speed
Plan 3.2.2.1.2: Speed: if too slow 3.2.2.1.2.1 if correct 3.2.2.1.2.2 if too fast 3.2.2.1.2.3
3.2.2.1.2.1 Increase speed
3.2.2.1.2.2 Maintain speed
3.2.2.1.2.3 Decrease speed

Fig. 3. Example of a sub-goal hierarchy extended onto a separate page.

From a reader’s perspective, to gain a complete grasp of the meaning of the representation of a sub-goal hierarchy, each plan has to be read and understood and then remembered in order to apply it to the set of sub-goals whose sequencing it describes. If plans are written in an abbreviated form, the meaning of the abbreviated form has to be checked if the reader is not familiar with the abbreviations used. In addition, the longer and more complex the plan, the greater the likelihood that some element may be forgotten, leading to the plan having to be re-read and the process repeated. It can be seen from Fig. 2 that the addition of plans to a sub-goal hierarchy makes the diagram more crowded and potentially more difficult to read. Whilst this could be addressed by simply spacing out the components of the diagram, there is then a tradeoff of readability against the proportion of the sub-goal hierarchy that can be represented on the page. This is of particular significance when undertaking the analysis of a large scale task where the sub-goal hierarchy alone might fill multiple pages, even without the inclusion of plans. This is illustrated in Fig. 4, which is a fragment of a sub-goal hierarchy for descending a Boeing 737 during an approach and landing at New Orleans airport [3]. The complete hierarchy fills two landscape US letter pages, without plan information being shown. Plans were written in a flowchart-style notation, as shown in Fig. 5, on a separate page.

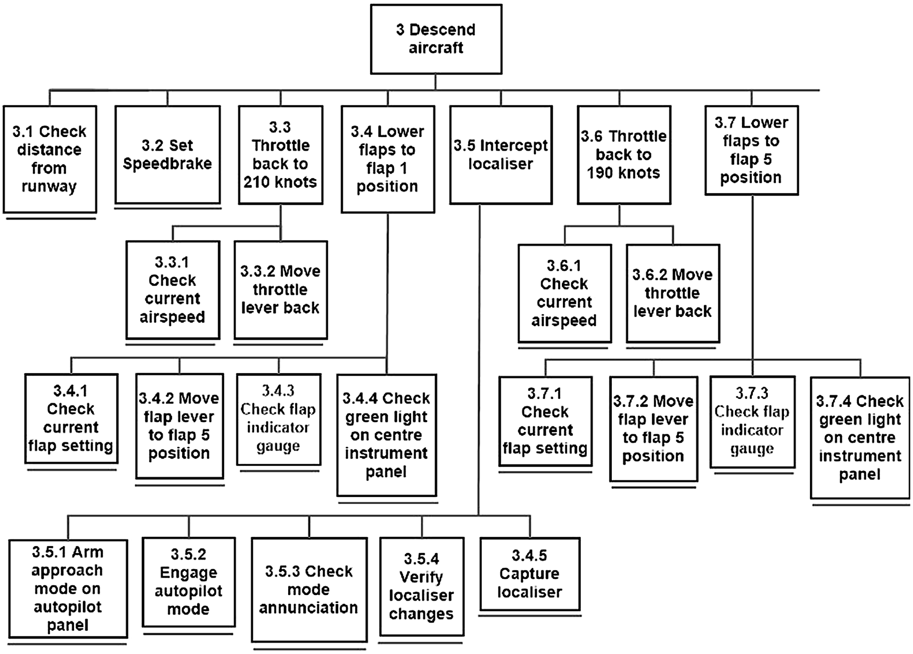

3 Descend aircraft
3.1 Check distance from runway
3.2 Set Speedbrake
3.3 Throttle back to 210 knots
3.4 Lower flaps to flap 1 position
3.5 Intercept localiser
3.6 Throttle back to 190 knots
3.7 Lower flaps to flap 5 position
3.3.1 Check current airspeed
3.3.2 Move throttle lever back
3.6.1 Check current airspeed
3.6.2 Move throttle lever back
3.4.1 Check current flap setting
3.4.2 Move flap lever to flap 5 position
3.4.3 Check flap indicator gauge
3.4.4 Check green light on centre instrument panel
3.7.1 Check current flap setting
3.7.2 Move flap lever to flap 5 position
3.7.3 Check flap indicator gauge
3.7.4 Check green light on centre instrument panel
3.5.1 Arm approach mode on autopilot panel
3.5.2 Engage autopilot mode
3.5.3 Check mode annunciation
3.5.4 Verify localiser changes
3.4.5 Capture localiser

Fig. 4. Fragment of a complex sub-goal hierarchy (adapted from Marshall et al. [3])

Separating the plans from the operations that they associated with can be problematic. Shepherd, when critiquing an early tabular format for representing HTAs, asserted that keeping plans in close proximity to their related operations would reduce confusion for the analyst and anyone reading the analysis [4]. The potential for confusion can be appreciated by reading plan 3 in Fig. 5 (deliberately placed on a separate page from Fig. 4) and then trying to remember it and apply it to the set of sub-goals to which it is related in Fig. 4. Part of the difficulty in doing this is that the sub-goal numbers in the plans have no implicit meaning. It is interesting to note that, if the numbers in the plans in Fig. 4 were to be replaced with the sub-goal boxes from Fig. 3, the result would be a set of hierarchical flow charts that completely describe the task.

There are a number of other features within Figs. 4 and 5 that are of interest. In Fig. 4, parts of the sub-goal hierarchy are repeated. The sub-goal hierarchy for sub-goal 3.3 is the same as that for sub-goal 3.6. By comparison, only one plan has been written, but there is a note on plan 3.3 which says use the same plan for sub-goal 3.6 with an appropriate value of airspeed. Space could be saved is the same concept of re-use was applied to the construction of sub-goal hierarchies. Also, the first decision point in plan 3.4.1 in Fig. 5 illustrates a branching plan. If the Flaps are not already at the required setting then the rest of the sequence is executed. However, if they are then no further action is taken. Finally, there is an issue with the use of underlines to indicate sub-goals that are not decomposed further, of which there are 20 in Fig. 4. From a reader's perspective, a key piece of information that has to be conveyed is if the abstraction hierarchy is continued on another page. This is indicated to the reader by the absence of

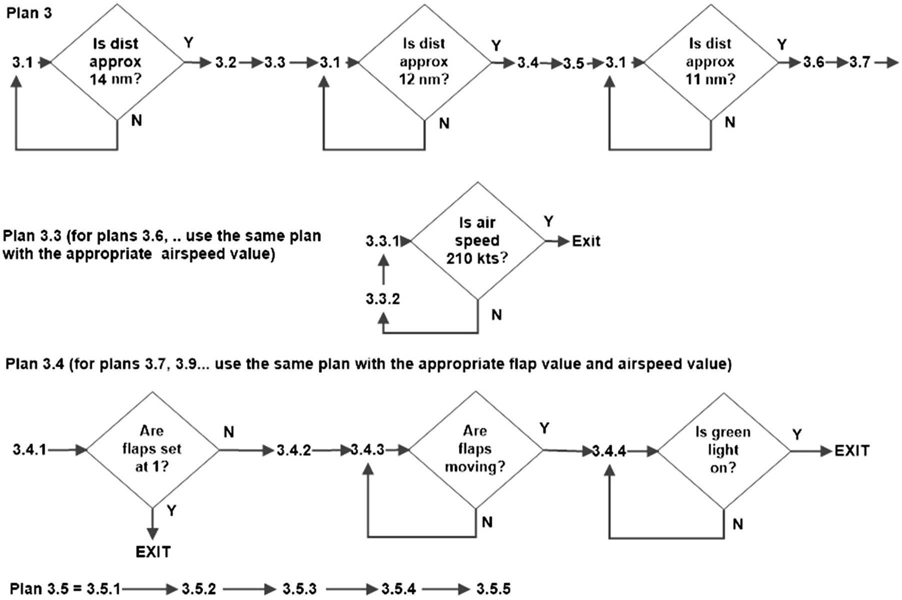


Fig. 5. Plans shown in flowchart notation (adapted from Marshall et al. [3])

an element of notation (no underline) rather than the presence of one (such as for sub-goal 3.2.2.1 in Fig. 2). If the notation indicated that a sub-goal was further decomposed elsewhere by exception, the underlines could be removed, reducing the visual clutter in the diagram. Twenty lines would be removed from Fig. 4.

3 New Notation

The notation presented in this section has been developed to address the issued identified with the presentation of sub-goal hierarchies and plans. The design objectives for the development of the notation were:

1. The notation should provide a clear, uncluttered representation of sub-goal hierarchies.
2. Sequencing of sub-goals should be implicit in the graphical representation.
3. Statements of triggering conditions should be proximal to the sub-goal(s) to which they apply.
4. The notation should explicitly identify if a sub-goal is further decomposed in a separate diagram (typically on another page).

The notation developed in this paper is derived from the diagrammatic notation used for representing hierarchical structures in software design. The Jackson Structured Programming method [5] employs a notation for describing data structures and designing program structures which uses the same hierarchical box/tree notation, but with some additions that denote repeated processes and selections structures. This notation has been extended in the Structured Systems Analysis and Design Methodology (SSADM) to also describe parallel actions, for use in modelling the life histories of entities in data processing systems [6].

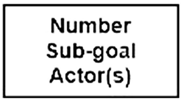
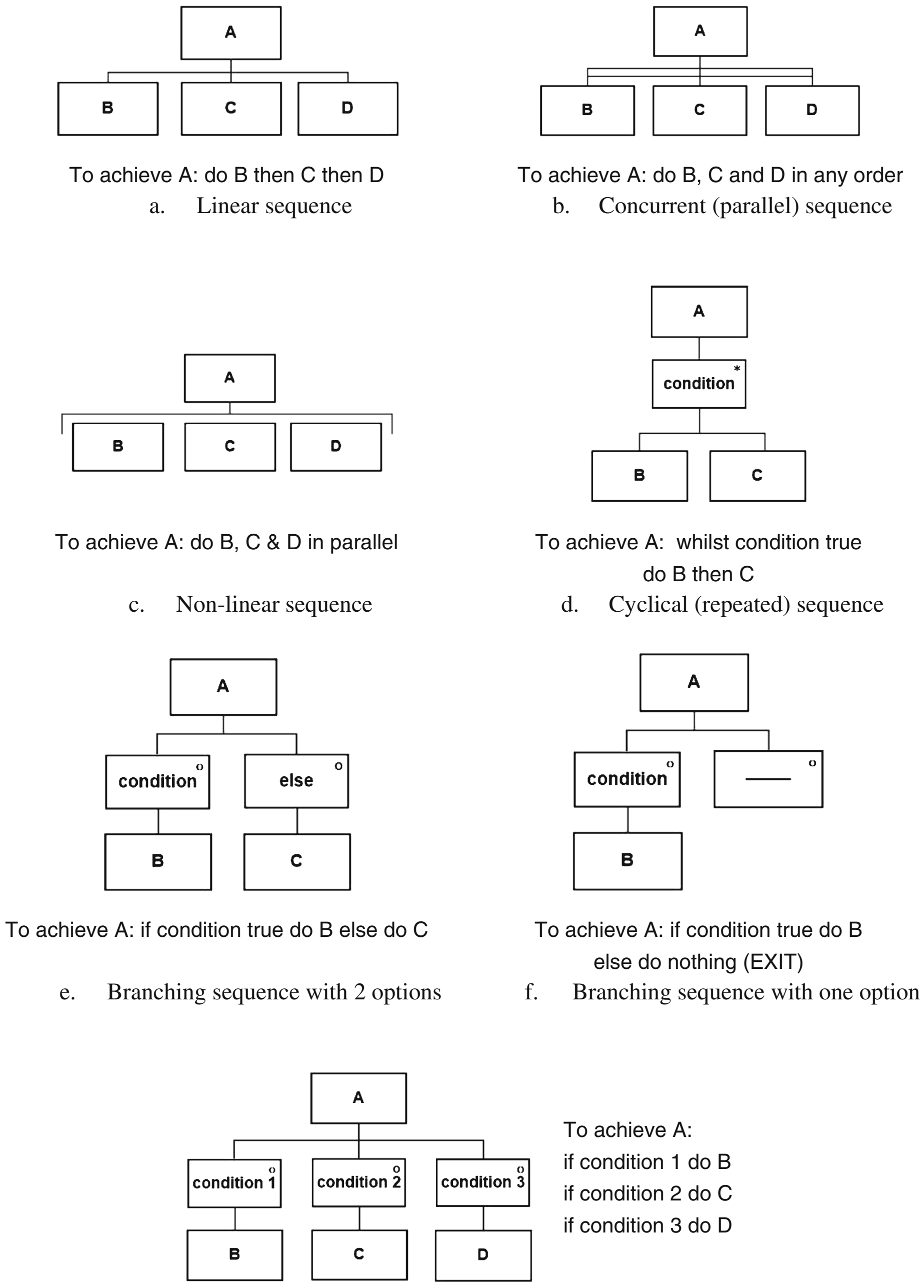


Fig. 6. Goal and sub-goal representation

Goals and sub-goals are represented by boxes in the same way as they are in conventional HTA notation. Each box is numbered for cross-referencing purposes and contains the sub-goal description. If a team task is being analyzed, it is useful to identify the actors(s) involved in achieving the sub-goal [7], so these may be included as required. The notation for representing sub-goal sequencing and triggering conditions is shown in Fig. 6. The text below each construct describes how it is read. Following Annett's observation that many tasks are proceduralised and so take the form of a linear sequence [1], the conventional HTA tree structure, with no other annotation, is used to represent a linear sequence (Fig. 6a) as the default representation. Any sequence which is not a linear sequence is indicated by an appropriate notational construct. Parallel horizontal lines linking sub-goals (Fig. 6b) are used to indicate that the operations described by the sub-goals are executed concurrently (in parallel). A horizontal bracket encompassing a group of sub-goals (Fig. 6c) indicates that the operations that they describe can be executed in any order (a non-linear sequence). The remaining notational constructs address sequences whereby the execution of operations described by the sub-goals is contingent on a condition being met. Figure 6d shows the representation for a cyclical sequence. A condition box is inserted between the superordinate sub-goal and the subordinate sub-goals that re-describe it. The multiplication sign in the top right hand corner of the condition box indicates that the set of sub-goals below it may be executed multiple times (and also indicates that it is not a sub-goal box). The condition statement captures the condition that has to be met for the execution of the subordinate sub-goals, such as "while in the cruise" (Fig. 7).

Figure 6e shows the notation for a branching sequence where, if a condition is met one set of operations is executed else an alternative set is executed. A condition box is inserted above each alternative set of sub-goals. The small circle in the top right-hand corner denotes that it is a selection condition. In situations where no action is required in the alternative case (such as "if hungry get food else do nothing"), the notation in Fig. 6f can be used. The horizontal line replacing the else statement in the second condition box indicates that no action is required. The final notational construct shown in Fig. 6g represents a selection sequence, in which a mutually exclusive set of conditions determine which set of operations are to be executed. These constructs can be combined to produce hybrid sequences as required.

Where a sub-goal, which is further decomposed, occurs in multiple locations in the sub-goal hierarchy (such as setting flaps in Fig. 4) the notation shown in Fig. 8 is used. This type of sub-goal construct is referred to in this notation as a procedure. It is



To achieve A: do B then C then D

a. Linear sequence

To achieve A: do B, C and D in any order

b. Concurrent (parallel) sequence

To achieve A: do B, C & D in parallel

c. Non-linear sequence

To achieve A: whilst condition true
do B then C

d. Cyclical (repeated) sequence

To achieve A: if condition true do B else do C

e. Branching sequence with 2 options

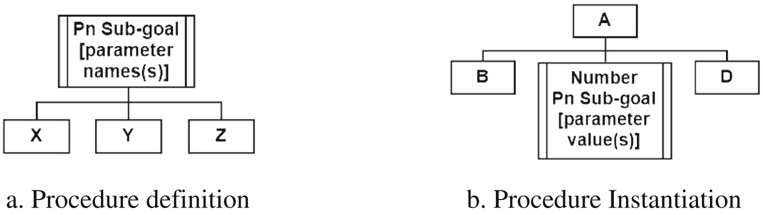
To achieve A: if condition true do B
else do nothing (EXIT)

f. Branching sequence with one option

To achieve A:
if condition 1 do B
if condition 2 do C
if condition 3 do D

g. Selection sequence

Fig. 7. Sequencing and triggering condition notation



a. Procedure definition

b. Procedure Instantiation

Fig. 8. Notation for procedure definition and instantiation

decomposed only once, using the notation shown in Fig. 8a. The vertical parallel lines indicate that it is a procedure and it is given a procedure number (Pn), which is used in the numbering of sub-goals in the procedure definition sub-goal hierarchy. If the operations in the procedure make reference to particular values (such as flap settings or speeds), which may be different each time a procedure is used, these values must be identified when the procedure is instantiated in the main sub-goal hierarchy, and a placeholder for them is required in the sub-goal hierarchy of the procedure definition. These values are referred to as parameters. They are named in the procedure definition (Fig. 8a) and the values specified in the procedure instantiation (Fig. 8b). Procedure definition and instantiation are illustrated fully in Figs. 12 – 13 in the next section.

The final notational construct is that for showing links between sub-goal hierarchy sections which may be shown on different pages. Numbered links are used, as shown in Fig. 9. For clarity, the sub-goal being decomposed is repeated in each section (sub-goals A and B in the examples in Fig. 9).

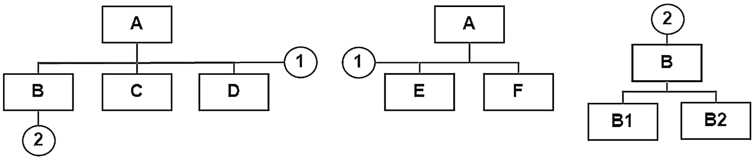


Fig. 9. Link notation

4 Example of the Use of the New Notation

Figures 10, 11, 12 and 13 illustrate the use of the new notation. Figure 10 shows the taxi to gate task shown in Fig. 2 recast in the new notation. The first line of decomposition shows how sequencing constructs can be mixed (parallel tasks occurring in an otherwise linear sequence). Cyclic sequences are also illustrated. Figure 11 shows the use of the selection construct. The use of procedures in Fig. 12 allows the Descend aircraft task sub-goal hierarchy, originally presented in Fig. 4, to be presented in a much more compact form, whilst also capturing the key relationship between sub-goals and the distance from the runway. Figure 13 illustrates procedure definition and the use of the branching construct. Figures 10, 11 and 13 also demonstrate that conditions boxes link sub-goals to their superordinate sub-goal by their triggering conditions, which is a literal application of Annett's analysis principles [1] discussed in Sect. 2.

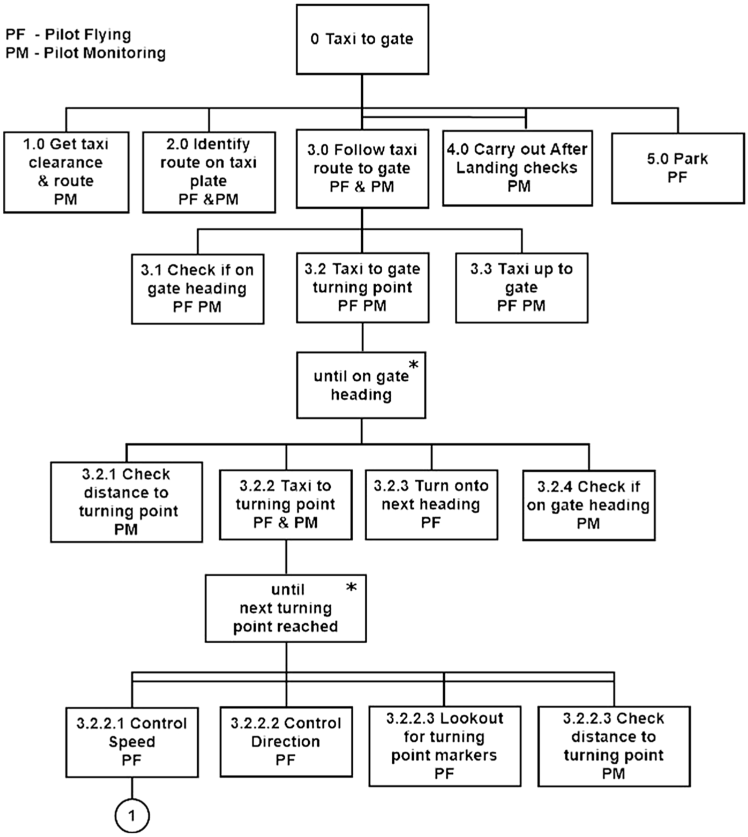

PF - Pilot Flying
PM - Pilot Monitoring
0 Taxi to gate
1.0 Get taxi clearance & route PM
2.0 Identify route on taxi plate PF &PM
3.0 Follow taxi route to gate PF & PM
4.0 Carry out After Landing checks PM
5.0 Park PF
3.1 Check if on gate heading PF PM
3.2 Taxi to gate turning point PF PM
3.3 Taxi up to gate PF PM
until on gate heading *
3.2.1 Check distance to turning point PM
3.2.2 Taxi to turning point PF & PM
3.2.3 Turn onto next heading PF
3.2.4 Check if on gate heading PM
until next turning point reached *
3.2.2.1 Control Speed PF
3.2.2.2 Control Direction PF
3.2.2.3 Lookout for turning point markers PF
3.2.2.3 Check distance to turning point PM
1

Fig. 10. Taxi to gate sub-goal hierarchy in new notation

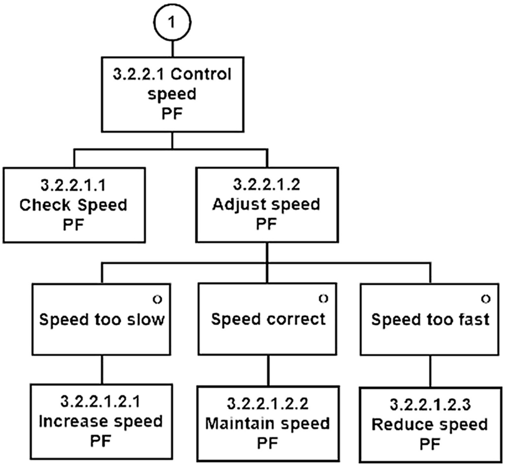

1
3.2.2.1 Control speed PF
3.2.2.1.1 Check Speed PF
3.2.2.1.2 Adjust speed PF
Speed too slow o
Speed correct o
Speed too fast o
3.2.2.1.2.1 Increase speed PF
3.2.2.1.2.2 Maintain speed PF
3.2.2.1.2.3 Reduce speed PF

Fig. 11. Control speed sub-goal hierarchy in new notation

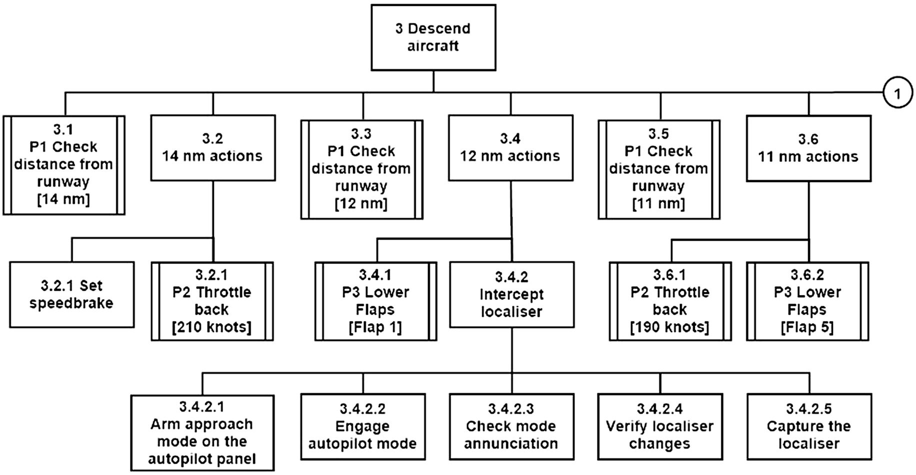

3 Descend aircraft
1
3.1 P1 Check distance from runway [14 nm]
3.2 14 nm actions
3.3 P1 Check distance from runway [12 nm]
3.4 12 nm actions
3.5 P1 Check distance from runway [11 nm]
3.6 11 nm actions
3.2.1 Set speedbrake
3.2.1 P2 Throttle back [210 knots]
3.4.1 P3 Lower Flaps [Flap 1]
3.4.2 Intercept localiser
3.6.1 P2 Throttle back [190 knots]
3.6.2 P3 Lower Flaps [Flap 5]
3.4.2.1 Arm approach mode on the autopilot panel
3.4.2.2 Engage autopilot mode
3.4.2.3 Check mode annunciation
3.4.2.4 Verify localiser changes
3.4.2.5 Capture the localiser

Fig. 12. Descend aircraft sub-goal hierarchy in new notation, illustrating the use of procedure instantiations.

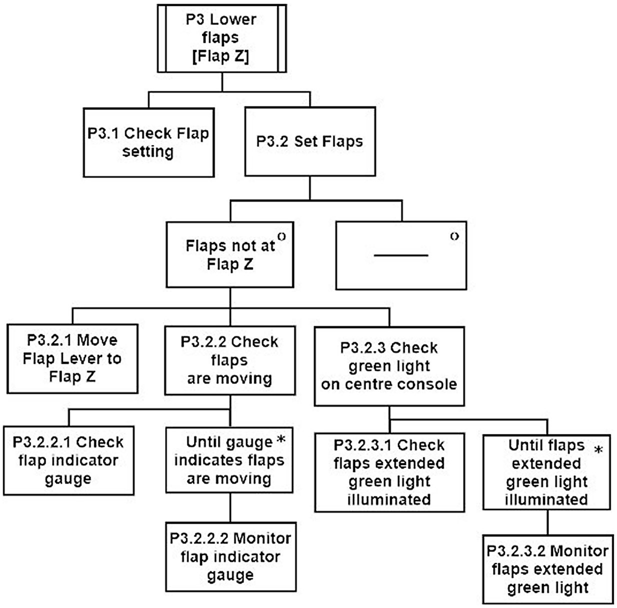

P3 Lower flaps [Flap Z]
P3.1 Check Flap setting
P3.2 Set Flaps
Flaps not at Flap Z
o
o
P3.2.1 Move Flap Lever to Flap Z
P3.2.2 Check flaps are moving
P3.2.3 Check green light on centre console
P3.2.2.1 Check flap indicator gauge
Until gauge indicates flaps are moving *
P3.2.3.1 Check flaps extended green light illuminated
Until flaps extended green light illuminated *
P3.2.2.2 Monitor flap indicator gauge
P3.2.3.2 Monitor flaps extended green light

Fig. 13. Lower Flaps sub-goal hierarchy shown as procedure definition in new notation

5 Conclusions

This paper has described and demonstrated the use of a new notation for representing hierarchical task decompositions that portrays graphically the information described in plans in conventional HTA notation. The human factors specialists and SMEs from the

Future Flight Deck Technologies project that have used the new notation to produce the examples shown in this paper consider that it yields sub-goal hierarchies that are both simpler both to produce and interpret, and that the notation meets its design criteria. However, further work is required to formally test these assertions.

Acknowledgement. This work was funded by Innovate UK as part of the Future Flight Deck Technologies project.

References

1. Annett, J., Duncan, K.D., Stammers, R.B., Gray, M.J.: Task Analysis. Department of Employment Training Information Paper 6. HMSO, London (1971)
2. Stanton, N.A.: Hierarchical task analysis: developments, applications, and extensions. Appl. Ergon. **37**, 55–79 (2006)
3. Marshall, A., Stanton, N., Young, M., Salmon, P., Harris, D., Demagalski, J.,Waldmann, T., Dekker, S.: 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 (2003)
4. Shepherd, A.: An improved tabular format for task analysis. Occup. Psychol. **49**, 93–104 (1976)
5. Jackson, M.: Principles of Program Design. Academic Press, London (1975)
6. Weaver, P., Lambrou, N., Wakeley, M.: Practical Systems Development using SSADM. Prentice Hall, London (2002)
7. Baber, C., Walker, G., Salmon, P., Stanton, N.A.: Observation study conducted at the fire service training college. Technical Report, Human Factors Integration Defence Technology Centre (2004)