Task Analysis at the Heart of Human–Computer Interaction

Dan Diaper
Middlesex University, UK

INTRODUCTION

The history of task analysis is nearly a century old, with its roots in the work of Gilbreth (1911) and Taylor (1912). Taylor's scientific management provided the theoretical basis for production-line manufacturing. The ancient manufacturing approach using craft skill involved an individual, or a small group, undertaking, from start to finish, many different operations so as to produce a single or small number of manufactured objects. Indeed, the craftsperson often made his or her own tools with which to make end products. Of course, with the growth of civilisation came specialisation, so that the carpenter did not fell the trees or the potter actually dig the clay, but still each craft involved many different operations by each person. Scientific management's novelty was the degree of specialisation it engendered: each person doing the same small number of things repeatedly.

Taylorism thus involved some large operation, subsequently called a task, that could be broken down into smaller operations, called subtasks. Task analysis came into being as the method that, according to Anderson, Carroll, Grudin, McGrew, and Scapin (1990), "refers to schemes for hierarchical decomposition of what people do." The definition of a task remains a "classic and under-addressed problem" (Diaper, 1989b). Tasks have been differently defined with respect to their scope: from the very large and complex, such as document production (Wilson, Barnard, & MacLean, 1986), to the very small, for example, tasks that "may involve only one or two activities which take less than a second to complete, for example, moving a cursor" (Johnson & Johnson, 1987). Rather than trying to define what is a task by size, Diaper's (1989b) alternative is borrowed from conversation analysis (Levinson, 1983). Diaper suggests that tasks always have well-defined starts and finishes, and clearly related activities in between. The advantage of such a definition is that it allows tasks to be interrupted or to be carried out in parallel.

Task analysis was always involved with the concept of work, and successful work is usually defined as achieving some goal. While initially applied to observable, physical work, as the field of ergonomics developed from World War II, the task concept was applied more widely to cover all types of work that "refocused attention on the information processing aspect of tasks and the role of the human operator as a controller, planner, diagnostician and problem solver in complex systems" (Annett & Stanton, 1998). With some notable exceptions discussed below, tasks are still generally defined with people as the agents that perform work. For example, Annett and Stanton defined task analysis as "[m]ethods of collecting, classifying and interpreting data on human performance."

BACKGROUND

Stanton (2004) suggests that "[s]implistically, most task analysis involves (1) identifying tasks, (2) collecting task data, (3) analyzing this data so that the tasks are understood, and then (4) producing a documented representation of the analyzed tasks (5) suitable for some engineering purpose." While there are many similar such simplistic descriptions, Stanton's five-item list provides an adequate description of the stages involved in task analysis, although the third and fourth are, in practice, usually combined. The following four subsections deal with them in more detail, but with two provisos. First, one should always start with Stanton's final item of establishing the purpose of undertaking a task analysis. Second, an iterative approach is always desirable because how tasks are performed is complicated.

The Purpose of a Task Analysis

Task analysis has many applications that have nothing to do with computer systems. Even when used in HCI (human-computer interaction), however, task analysis can contribute to all the stages of the software-development life cycle. In addition, task analysis can make major contributions to other elements associated with software development, in particular the preparation of user-support systems such as manuals and help systems, and for training, which was the original application of hierarchical task analysis (HTA; Annett & Duncan, 1967; Annett, Duncan, Stammers, & Gray, 1971). HTA was the first method that attempted to model some of the psychology of people performing tasks.

Although infrequently documented, identifying the purposes for using task analysis in a software project must be the first step (Diaper, 1989a) because this will determine the task selection, the method to be used, the nature of the outputs, and the level of analysis detail necessary. The latter is vital because too much detailed data that does not subsequently contribute to a project will have been expensive to collect, and too high a level will require further iterations to allow more detailed analysis (Diaper, 1989b, 2004). Decomposition-orientated methods such as HTA partially overcome the level-of-detail problem, but at the expense of collecting more task data during analysis. Collecting task data is often an expensive business, and access to the relevant people is not always easy (Coronado & Casey, 2004; Degen & Pedell, 2004; Greenberg, 2004). Within a software-development life cycle, Diaper (2004) has suggested that one identify all the stages to which a task analysis will contribute and then make selections on the basis of where its contribution will be greatest.

Identifying Tasks

In the context of task scenarios, which Diaper (2002a, 2002b) describes as "low fidelity task simulations," Carroll (2000) rightly points out that "there is an infinity of possible usage scenarios." Thus, only a sample of tasks can be analysed. The tasks chosen will depend on the task analysis' purpose. For new systems, one usually starts with typical tasks. For existing systems and well-developed prototypes, one is more likely to be concerned with complex and difficult tasks, and important and critical ones, and, when a system is in use, tasks during which failures or problems have occurred. Wong (2004) describes his critical decision method as one way of dealing with the latter types of tasks.

Unless there are overriding constraints within a software project, then task analysts should expect, and accept, the need to be iterative and repeatedly select more tasks for analysis. Since the coverage of all possible tasks can rarely be complete, there is a need for a systematic task selection approach. There are two issues of coverage: first, the range of tasks selected, and second, the range of different ways that tasks may be carried out, both successfully and unsuccessfully.

One criticism of task analysis is that it requires extant tasks. On the other hand, all tasks subjected to task analysis are only simulations as, even when observed in situ, a Hiesenberg effect (Diaper, 1989b) can occur whereby the act of observation changes the task. Often, it is desirable to simulate tasks so that unusual, exceptional, and/or important task instances can be studied and, of course, when a new system or prototype is not available.

Collecting Task Data

There are many myths about task analysis (Diaper et al., 2003), and one of the most persistent involves the detailed observation of people performing tasks. Sometimes, task-analysis data do involve such observation, but they need not, and often it is inappropriate even with an existing system and experienced users.

Johnson, Diaper, and Long (1984; see also Diaper, 1989b, 2001) claim that one of the major strengths traditionally associated with task analysis is its capability to integrate different data types collected using different methods. The critical concept is that of fidelity. According to Diaper (2002a, 2002b), "fidelity, a close synonym is validity, is the degree of mapping that exists between the real world and the world modelled by the (task) simulation," although as he says parenthetically, "N.B. slightly more accurately perhaps, from a solipsistic position, it is the mapping between one model of the assumed real world and another."

At one end of the task-fidelity spectrum there is careful, detailed task observation, and at the other,