

## FROM TASK ANALYSIS TO USER INTERFACE DESIGN

François Aubin<sup>1</sup>, Jean-Marc Robert<sup>2</sup>, Daniel Engelberg<sup>1</sup>

<sup>1</sup> Centre de Recherche Informatique de Montréal (CRIM), 1801 McGill College, Bureau 800, Montréal, Québec H3A 2N4

<sup>2</sup> École Polytechnique de Montréal, Département de génie industriel, C.P. 6079, Succ. centre-ville, Montréal, Québec H3C 3A7

### Abstract

Currently even the most structured approaches to interface design involve a substantial amount of intuition and individual judgment. This paper presents a method for creating a more structured approach by mapping task analysis directly to user interface design. This method has the potential to accelerate, standardize, and eventually automate some parts of the interface development cycle.

### Introduction

In order to provide greater structure and consistency to the early stages of the design process, we propose a method which directly incorporates human factors guidelines into a task analysis model based on Hierarchical Task Analysis (HTA). This augmentation of the HTA model will allow us to allocate tasks between the human and computer, and to map the task model directly onto an interface design.

Most usability engineers emphasize the importance of task analysis in the usability engineering life cycle. GOMS<sup>1</sup>(Card et al., 1983), one of the most often cited methods for task analysis in human-computer interaction literature, is valuable for establishing the user's goals and methods, and for predicting human performance measures such as execution time. But taken alone, GOMS does not provide enough information to specify an interface design.

Our proposal extends the HTA approach by linking task operators with human factors principles and guidelines, interface objects and interaction techniques. Task operators map to interaction techniques (e.g. click the left mouse button) or spatial layouts (e.g. bar chart, table, network.) via human factors design principles and guidelines.

Our current set of task operators, with their associated interface object guidelines, is listed in Table 1. Guidelines are drawn from the engineering psychology literature (e.g. Wickens, 1992).

These task operators have been developed empirically over the course of eight commercial projects in domains such as banking, customer service, credit investigation, and finance. The list does not cover other domains, nor does it form a comprehensive taxonomy of task operators. However we have found it highly effective in that it has allowed us to describe user tasks in most of the cases we encountered.

**Table 1. Task operators and their interface object guidelines.**

<b>Task Operator</b>	<b>Definition</b>	<b>Examples of guidelines for interface object</b>
compare (approximate)	Examine two or more objects in order to discover rough similarities and differences	For <b>quantitative</b> values, best results are obtained with two linear scales aligned with the same baseline.
compare	Examine two or more objects in order to	For <b>quantitative</b> values, text is superior

---

<sup>1</sup> Goals, Operators, Methods, Selection rules

(exact)	discover precise similarities and differences	to graphics.
discriminate	Examine two or more objects in order to discover differences.	For <b>graphic</b> objects, graphics are superior to text.
recall	Bring an object from long-term memory back to awareness.	For <b>all types</b> of objects, automate the task if possible.
recognize	Relate a perceived object to memories of that object or class of objects	For <b>abstract</b> objects, text is superior to graphics.
scan	Survey a complex object by glancing over all of its elements	For <b>quantitative</b> values, orient items in a list or a table.
detect	Discover presence of an object or a property	For <b>graphic</b> objects, automate the task if there is no noise in the signal.
calculate	Perform a mathematical operation on two or more objects	For <b>quantitative</b> values, automate the task.
select	Choose from among a number of objects	For <b>all types</b> of objects, use written menu with short-cuts.
enter (text)	Enter text into the system	For <b>text</b> , use keyboard.
integrate	Synthesize several objects into a functioning whole	For <b>graphic</b> objects, display the objects and use direct manipulation.
correlate	Evaluate similarities in trends of two or more objects.	For <b>quantitative</b> values, automate the task.
judge	Form an opinion through careful weighing of evidence.	For <b>all types</b> of objects, allocate the task to humans.
decide	Arrive at a solution that ends uncertainty.	For <b>all types</b> of objects, present cues simultaneously in order to avoid bias. If there is uncertainty, allocate the task to humans.

#### Criteria for identifying task operator

1. The task operator must relate to the cognitive goal of the user.
2. Task operators must be tool-independent. For example, “click the left mouse button” is not an acceptable operator. Instead, we would use “select” as the operator and the human factors recommendation will refer us to “click the left mouse button”.

#### Task optimization and allocation

Task optimization consists of eliminating unnecessary operations (e.g. repeated entries), adding new functionality and reorganizing the task. The task allocation consists of allocating functions between human and machine. It is important to do task optimization and allocation before task mapping in order to avoid mapping task that will be eliminated.

#### High-level design of the interface

The goal of the high-level design is to define the main screens, the storyboard (Mayhew, 1992), and the main manipulations, and to select interaction styles according to the technical constraint and the user profile. To do so we define empirically three general principles for solving the most common problems:

1. Each screen must contain only and all the information required to do a task.
2. Organization of screens must be compatible with the task structure.
3. Organize screens according to priority and frequency of use; secondary tasks must be presented in secondary windows or subscreens.

Empirical testing can start immediately after the high-level design is performed.

#### Detailed design

Detailed design specifies spatial layout of information within a screen, and the nature of each message and interaction technique. The mapping process establishes a relationship between task operator, object, guideline, dialogue specification and interaction technique. Since the screens and windows are compatible with the task structure (rule 2 of high-level design), detailed mapping allows us to specify the lowest (most concrete) level of the task hierarchy.

The mapping principle for solving the most common problems are the following:

1. **Interface behavior** (e.g. displaying warning messages) is determined by the task operators, task objects, human factors guidelines, and domain-specific standards (e.g. commercial practices).
2. **Implementation of interface behavior** (i.e. interface objects, layout and interaction techniques) is determined by the interface behavior (1) and the standards of the operating environment.
3. High-level design rules 2 and 3 (q.v.) are also applied.

### Example

In the context of the customer service department of a utility company, users must answer customer queries and satisfy the customer. Figure 1 shows the high level design of a screen. Figure 2 shows a detailed mapping of the task “Check payments” to the interface design. Note that high-level task operators such as “check” must be mapped to one or several of the lower-level task operators defined in Table 1.

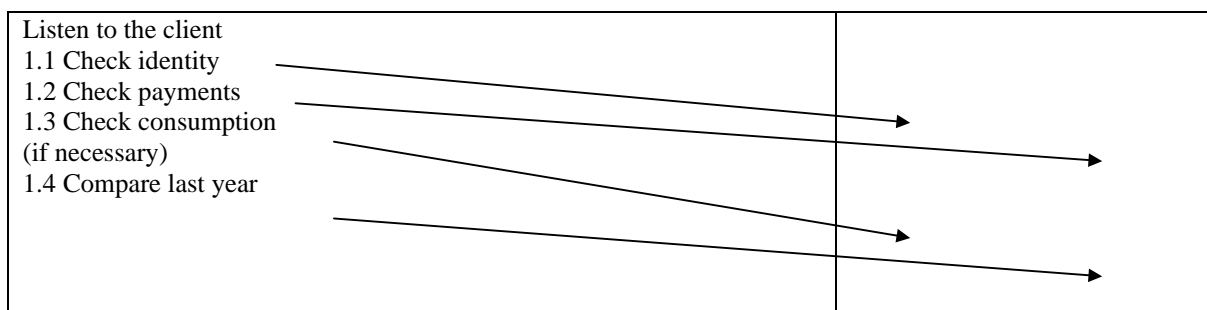
### Discussion

This formal method is practical and is currently being applied in various commercial projects. Its main advantage is to link human factors guidelines to the task model. The extension of this method to new content domains is the object of a larger research project. We are currently working on the extension, classification, and testing of new operators, and their matching with relevant human factors guidelines and interaction techniques.

### References

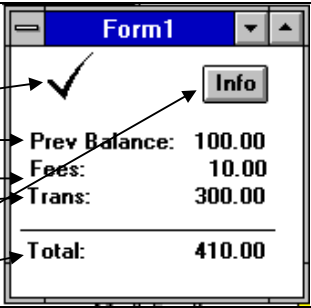
- Card, S.K., Moran T.P., and Newell, A. 1983, *The Psychology of Human-Computer-Interaction*, Lawrence Erlbaum Associates, Hillsdale, NJ.
- Mayhew, D.J. 1992, *Principles and Guidelines in Software User Interface Design*, Prentice Hall, New Jersey.
- Wickens, C.D. 1992, *Engineering Psychology and Human Performance*, Harper Collins, New York.

### Figures



**Figure 1.** High-level design.

Task operator- <i>object</i>	Interface behavior	Spatial layout	Screen
------------------------------	--------------------	----------------	--------

<p>Discriminate- <u>agreement</u>  Recognize- <u>previous balance</u>  Recognize- <u>fees</u>  Recognize- <u>transaction</u>  Select- <u>info on origin</u>  Recognize- <u>total</u></p>	<p>If apply, visible  Fixed<sup>2</sup>, if=0, no  Fixed, if=0, no  Fixed, if=0, no  Disable if no  Fixed, if=0, no</p>	 <p>Icon  Text  Text  Text  Button  Text</p>
--	---	---

**Figure 2.** Detailed design using task operators defined in Table 1.

<sup>2</sup> *Fixed*: user cannot modify the object