Introduction: Choosing Functionality with a High-Level GOMS Tasks Analysis

Definitions

Task Analysis Methods

Problems in the Design of Functionality

Reprise: When to do GOMS Modeling: During Design

Using a High-Level GOMS Model Representation of a Task Analysis in the Design of Functionality

High-Level GOMS Models

Task Analysis and the Design of Functionality

David E. Kieras
University of Michigan

Used with permission and adapted by Anthony J. Hornof, University of Oregon

Additional Acknowledgement: Ruven Brooks, Rockwell Automation

Definitions

Task Analysis: Understanding the user’s task well enough to help design a system that will effectively support the users in doing the task.

Task: The user’s job or work activities; what the user is trying to accomplish.

Analysis: A relatively systematic approach to developing and documenting what is involved in the tasks.

Not just unaided intuition or speculation.

Design of functionality: Choosing the user-accessible functions of a system - what functions will the system provide?

Concept:

Do a task analysis prior to designing the system.

Choose functionality based on task analysis.

Help ensure that system will actually be useful.

Augment the task analysis after the system is designed.

Find out how the system will actually get used.

Related to user testing and analytic evaluation, but the goal is to understand and describe the user’s task, not to evaluate an interface design.

Evaluation can result from the task analysis, but a task analysis can be done before there is any system design has been proposed!

First, collect data about the task, either with existing or proposed system.

Observe/interview/shadow users, ethnographic studies, videos, computer usage logs, time-use diaries, talk to designers, examine existing interfaces.

What does system and user do, and how do they do it?

Second, represent results of the task analysis in a notation that aids understanding and interpretation.

What does the whole system do?

Where does the user fit in? What role does the user play?

What specific tasks does the user perform?

• What does the user need to know?

• What does the user need to do?

• What does the user see and interact with?

• What can the user do wrong?

Many different task analysis methodologies developed by Human Factors field.

Applied to a broad range of systems and interfaces.

Single-person simple devices

• E.g. hand-held radio.

Multiperson, multimachine systems.

• E.g. Warship combat information center.

Methodologies provide:

Guidance on what to look for when doing the task analysis.

Notations for recording the results of a task analysis.

Information about either existing or proposed systems.

Help identifying usability problems along the way.

But methodologies do not “do” the task analysis for you.

Task analysis is inherently an informal, heuristic process.

Analyst must decide what to look for, what to record, and how to interpret the results.

Choose a methodology that is helpful for the design problem at hand.

Problems in the Design of Functionality

Usual design process: Choose functionality, then design user interface.
   Functionality often chosen haphazardly, independently of usability.

But choice of functionality is most important factor in usability.
   A superior interface can’t compensate for an inadequate set of functions!
   Examples:
   • A word processor with no footnote function - no way to make footnoting easy in the interface!
   • First generation digital diaries - no time/date or alarm functions!

Indirect damage due to poor choice of functionality: Divert resources to useless functions.
   Need to know what functions are important!
   Examples:
   • Database which after usability problems were corrected turned out to be unnecessary.
   • Word processor with useless multiple-column functionality.

Functions need to be chosen based on task analysis.
   Provide those functions that result in a product that is both useful and usable.

Problem: How to combine task analysis, choice of functions, and usability?
   Just guess, then see how it turns out?

Reprise: When to do GOMS Modeling: During Design

Carry out GOMS analysis as design decisions are made
   Same analytic methods, just difference in when done
   Design decisions known, rather than reconstructed or inferred

Bad parts of design space avoided from the start
   First design is likely to be fairly good
   Bad choices (e.g. complex methods) immediately revealed
   Informal analysis may suffice to check decisions
   • E.g., can immediately tell messy methods will result

Using a High-Level GOMS Model Representation of a Task Analysis in the Design of Functionality

Proposed solution: Choose functionality based on high-level GOMS analysis of what is required in the user’s tasks

1. Write out high-level methods for user’s top-level goals
   • Methods should be as independent as possible of specifics of user interface design
   • If done properly, can be used as the top level of final GOMS model of the interface

2. Choose functions that allow goals to be accomplished with simple high-level methods.
   • Ensures that system will be both useful and at least reasonable usable.

3. Choose interface and hardware that allows methods to be simple and meets other requirements.
   • E.g. cost, performance, what platforms are available, what interface users are accustomed to.

4. Elaborate design by writing lower-level methods.
   • Ensure interface to chosen functions is highly usable.

5. Revise functionality, methods, interface choices as needed

A process to use at the beginning of relatively standard development processes
   A more precise, usability-oriented, form of conventional requirements analysis.

High-Level GOMS Models

A GOMS model, but at a level above specific interface methods

Goals
   Refer only to non-interface aspects of user’s task.

Operators
   High-level mental operators
   • Think-of operators
   • Decide operators
   Invocations of system functions
   • Unanalyzed high-level operators
   • E.g., update the database
   • May not be specific interface interactions
   • Much higher-level than keystroke
   • RIGHT: update the database
   • WRONG: click on UPDATE button

Methods
   Show order in which mental operators and system functions are executed.

The model should include:
   What information users need for system functions and decisions.
   • Where does the user get it?
   How users will detect and correct errors
   • When should the user verify the situation?
   • When can errors happen?
   • Incorrect results from mental operators
   • Slips in function invocation
   • What information is needed to detect an error?
   • How can the user correct an error?
   • At least a place-holder; specific interface decisions might be involved
The start of a GOMS analysis for the design of the functionality needed to get directions from the web using a Palm handheld with a wireless modem

By Anthony Hornof - 5/3/01

Derived in part from Palm Computing “Guide to Web Clipping Applications” P/N: 406-2391

Sample Task

• Loretta and Brenda are visiting NYC, but they don’t know it too well. They just left a meeting in Greenwich Village (at 51 Astor Place) and need to get to a meeting uptown (at 945 Madison Ave.)
• For some reason, they are driving.
• They need directions.

High-Level Methods

Method for goal: Drive to a new place.
Step 1. Accomplish Goal: Get directions.
Step 2. Accomplish goal: Drive there.
Step 3. Return with goal accomplished.

Selection rule set for goal: Get directions.
If you want to look like a tourist: Ask someone for directions.
If you have a map: Find a route on the map.
If you have a handheld Palm with wireless web access: Get directions using the MapQuest Web Clipping App.

Assumptions

• Loretta has a Palm handheld computer with a wireless modem and a web browser, and the MapQuest Web Clipping App.
• Loretta is an experienced user of the system.
• Everything works.
• Brenda is driving.

Question:
What are the basic steps that are going to be required to get directions from the internet using a modem on a Palm handheld device?

Web Clipping

A Web Clipping application interactively retrieves small bits of information from the Internet via a wireless transmission, such as the nearest Starbucks Coffee shop, flight schedules, or driving directions.

On a desktop computer browser, there is lots of space for graphics, links, advertisements, white space:

But not on small handheld device:

![MapQuest Web Clipping App](image)
User Goals

MFG: Get directions using the MapQuest Web Clipping App.
Step 1. AG: Connect to network.
Step 2. AG: Start up the MapQuest Web Clipping App.
Step 3. AG: Download directions using MapQuest.
Step 4. RGA.

- MFG: Download directions using MapQuest.
  - Step 1. AG: Enter the “from” address.
  - Step 2. AG: Enter the “to” address.
  - Step 3. AG: Initiate download.
  - Step 4. Verify the directions are downloaded.
  - Step 5. RGA.

Choose functions, screen layouts, and user feedback that allow goals to be accomplished.

Choose functions and screen layouts that allow goals to be accomplished.

Choose functions, screen layouts, and user feedback that allow goals to be accomplished.

Provide the user with the feedback necessary to accomplish each subgoal.

Figure 1. Screenshot of the input fields used by goals “Enter an address.” From the MapQuest Web Clipping Application.

To submit a query to the Internet:

(The procedures provided by the Web Clipping Documentation)

1. Tap the icon of the web clipping application you want to open.
2. Follow the directions onscreen for specifying the information you want from the Internet.
3. Tap the button or other object that displays the over-the-air icon to submit your query to the Internet.
   - The response — your clipping— typically appears onscreen within seconds.

Choose functions and screen layouts that allow goals to be accomplished.

Choose functions and screen layouts that allow goals to be accomplished.

Elaborate and Revise.
To find driving directions:

(From the Web Clipping Documentation)

1. Tap the Applications icon.
2. Tap the MapQuest icon.
3. Under From, tap the Address edit line and enter the street address of the place you are leaving from.
   You can use Graffiti writing or the onscreen keyboard to enter data.
4. Complete the From address by entering the city and state.
5. Under To, tap the Address line and enter the street address of the place you want to go to.
6. Complete the To address by entering the city and state.

Directions from MapQuest.com
1: Start out going Southeast on E 8TH ST towards ASTOR PL.
2: E 8TH ST becomes ASTOR PL.
3: Turn LEFT onto 3RD AVE.
4: Turn RIGHT onto E 14TH ST.
5: Turn LEFT onto 1ST AVE.
6: 1ST AVE becomes 1ST AVE/UNITED NATIONS PLZ.
7: 1ST AVE/UNITED NATIONS PLZ becomes 1ST AVE.
8: Turn LEFT onto E 57TH ST.
9: Turn RIGHT onto PARK AVE.
10: Turn LEFT onto E 73RD ST.
11: Turn RIGHT onto MADISON AVE.

TOTAL ESTIMATED TIME: 10 minutes
TOTAL DISTANCE: 4.1 miles (6.7 km)

Niether set of directions are perfect.

Using a High-Level GOMS Model
Representation of a Task Analysis in the Design of Functionality

Proposed solution: Choose functionality based on high-level GOMS analysis of what is required in the user’s tasks

1. Write out high-level methods for user’s top-level goals
   - Methods should be as independent as possible of specifics of user interface design
   - If done properly, can be used as the top level of final GOMS model of the interface

2. Choose functions that allow goals to be accomplished with simple high-level methods.

3. Choose interface and hardware that allows methods to be simple.
   - In the MapQuest example, we have identified the most fundamental exchanges of information necessary to accomplish the task.
   - So far the structure is straightforward, but it would have been easy to make any of these goals, or any of the further subgoals, more difficult than necessary.
   - We still have to design the functionality, displays, and controls.

4. Elaborate design by writing lower-level methods.
   - Ensure interface to chosen functions is highly usable.

5. Revise functionality, methods, interface choices as needed.