A TAXONOMY OF USER INTERFACE TERMINOLOGY

MARK H. CHIGNELL

Abstract

User interface design and analysis is an inherently interdisciplinary activity that merges cognitive, computing, and engineering sciences. Due to the rapid pace of technological change, there is as yet no science of human-computer interaction and little consensus on what the core knowledge of the discipline should be. In other sciences, the development of taxonomies, such as the taxonomy of living organisms in biology, has proved to be a useful foundation for scientific activity. This paper proposes a taxonomy of user interface terminology as a possible basis for the eventual development of human-computer interaction as a science. This taxonomy includes a model of the basic components of the interface and coverage of some of the major cognitive engineering principles that form the basis for human-computer interaction.

Introduction

The fields of human-computer interaction and user interface design are potentially vast, drawing on the findings of the social sciences, computer science, and empirical studies of how people react to the user interfaces that they are provided with. One of the most annoying things for user interface researchers is the feeling that one cannot keep up with this broad literature of things that one should know about. Drawing on psychological theory, it seems that the task of the researcher might be simplified somewhat by providing a framework or organizing schema within which to understand and absorb the frightening amount of possibly relevant material that should be dealt with.

The development of taxonomies or hierarchical structurings of domain-relevant terms, such as the taxonomy of living organisms in biology, has proved to be a useful framework for scientific activity in other disciplines. This paper proposes a taxonomy of user interface terminology as a possible basis for the eventual development of human-computer interaction as a science. Since this taxonomy currently represents the views of a small sample of researchers, it is necessarily idiosyncratic and will benefit from further revision and analysis. However, there can be no doubt that an appropriate taxonomy of terms would be of great assistance to human-computer interaction and education. Such a taxonomy could serve as a standard structure for archiving and discussing information relevant to user interfaces. It is envisioned that this taxonomy will eventually form the basis for a hypermedia-based research database environment that will allow user interface research groups and instructors to catalog their own literature reviews of important topics.

Original Motivation and Construction Method

The motivation for the current version of the taxonomy grew out of the demands of teaching a graduate course on human-computer interaction over a number of years. The first thing that one notices in teaching such a course is that there is no one text that covers the discipline. In fact, it is extremely difficult even to decide what topics should be covered in a one semester course. Part of the solution to this dilemma was to create a two-semester sequence of courses, one dealing with the theoretical foundation of human- computer interaction (cognitive engineering), the other dealing with tools and skills such as task analysis and rapid prototyping that can be used to develop advanced interfaces (intelligent interfaces). However, segmenting the field in this way did not solve the problem of identifying appropriate texts. It seemed that each text covered only a portion of the relevant issues of the field, and that the rapid progress in user interface technology tended to make books obselete in short time even if they did provide good coverage.

This problem is reflected in the different texts that have been used in the course in the past few years. These texts include Norman and Draper (1986), Carroll (1987), Baecker and Buxton (1987) and Stillings, Feinstein, Garfield, Rissland, Rosenbaum, Weisler, and Baker-Ward (1987). Other texts considered or referred to included Shneiderman (1987), Helander (1988), and Bailey (1988). Each of these texts provides a useful insight into important aspects of human-computer interaction and cognitive engineering, but none of them provide a coherent discussion of all the major topics relevant to human-computer interaction. Of course, it might be argued that the topic is simply too broad and diverse to expect adequate coverage in a single text, yet that did not explain the overall feeling of unstructuredness that one got about the discipline after reading the various texts.

Like many other professors who could not find an adequate text for their course, I began to contemplate writing a text myself. Yet this required selection and organization of topics. Thus the goal was formulated of developing a taxonomy for human-computer interaction that would then provide a basis for research and instruction.

The development of the taxonomy began with the assumption that the relevant topics could be organized into a hierarchical representation. While a tangled network might ultimately be more appropriate the strategy was to start by forcing everything into a tree structure and then create associative links across the branches of the tree as found necessary.

The first step in developing the hierarchy was the selection of a relevant set of terms that spanned the broad topic of human-computer interaction. This in itself was no easy task. Eventually, the terms were chosen from three sources: section headings in the handbook of human-computer interaction (Helander, 1988); index terms from a well-known book on user interface design (Shneiderman, 1987); lecture notes from the courses on cognitive engineering and intelligent interfaces that were taught at the University of Southern California. The terms that were initially selected have been further refined based on the comments of various colleagues and reviewers.

A sorting task (carried out by the author) was then used to organize the selected terms into a hierarchy. The terms were each written on separate 3 x 5 cards. The cards were then shuffled and spread over a large table top. The cards were then grouped together according to which cards appeared to belong together. This was done without any (conscious) preconceived notion of what the organization should look like. Once each group was formed it was collected together (tied with a rubber band) and then assigned a label. After all the cards had been placed in groups, the groups themselves were formed into groups. This process was continued until all the subgroups (and the terms that they contained) had been collected into a single supergroup representing the root of the tree.

The results of the sorting process were then transcribed onto a large sheet of paper and edited. This editing included revising the names of terms and categories to fit in with standard usage, and the expansion of terms within a group. This type of expansion was found to be much easier after a basic structure (taxonomy) had been defined. The revised taxonomy was then passed to members of the user interface and hypermedia groups at the Institute of Systems Science, National University of Singapore, for their review. Further revisions then led to the current version of the taxonomy.

This taxonomy is designed to organize terms in human-computer interaction for use in research and instruction. Thus its aim and character is different from the taxonomy of user-oriented functions developed earlier (Carter, 1986). Carter's taxonomy was concerned with the standardization of user interface functions or activities. In this taxonomy, we are concerned with the standardization of terminology over the discipline of human-computer interaction. Our interpretation of human-computer interaction is broadly defined, ranging from cognitive science issues underlying user behaviour to design guidelines and tools.

The Four Main Categories of Terms

The taxonomy that resulted from the process described above had four main branches, which are described in this section. A detailed analysis of each of these branches will be given in the following section.

The four main branches (categories of terms) of the taxonomy are as follows:

1. The Basic Interface Model
2. Cognitive Engineering
3. User Interface Engineering
4. Applications

The basic interface model is a simple characterization of the major components of all interfaces. In this model, a transaction begins with a (task-related) goal in the mind of the user. This then leads to a user behaviour (such as pointing with a pointing device and similar behaviours) which occurs in the context defined by the current status of the task and the computer system. The behaviour and the context in which it occurs then jointly define an action that is carried out by the system (such as retrieving a file, or presenting requested information on the screen). Displays consist of information that is presented to the user for its own sake, i.e, to be read or listened to. Effects are outputs from the system that are designed to assist the user in interpreting the system's actions and updating their mental model of system and task status. Forms are consistent models in which actions, effects, and displays are embedded. They generally conform to the notion of interface metaphors.

The basic interface model contains declarative knowledge about what user interfaces consist of. The second main category is cognitive engineering, which covers the areas of applied cognitive science that are relevant to understanding human-computer interaction.

The user interface engineering category includes subtopics that are relevant to the interface engineering process, without making any commitments as to what the (procedural) details of that process might be. While there have been numerous books and papers on the software engineering process, and on processes of design in general, the user interface design and implementation process is much less well defined. For instance, the overall process of user interface design is not directly addressed by Hartson and Hix (1989) in their lengthy review of human-computer interface development.

The fourth main category deals with general classes of application to which user interface design is directed. It is difficult to discuss broad issues in user interface design without regard to the type of application being considered. In addition, the type of interface often has a strong influence on the type of user interface that is developed and the general issues that are considered. For instance, user navigation issues tend to predominate in information technology interfaces, while cursor movement and text selection tends to be particularly important in text editing applications.

Each of the four main branches has potential controversies attached to it. For instance, there may be other ways of defining a basic interface model, or other topics in applied cognitive science that should take precedence over the topics selected here. However, while the details of each branch may be debatable, it seems useful to distinguish between the broad issues represented by each branch. The major impact of this taxonomy is that it distinguishes between the basic interface model (the what) and the interface engineering process (the how), and it also recognizes the importance of cognitive engineering as a foundation for human-computer interaction. In addition, the heterogeneity of user interfaces accross different types of application is also recognized.

Detailed Classification of Terms

The four main categories in the taxonomy may be further subdivided into branches and sub-branches. As we increase the level of detail, there is more chance that the micro-structure of the taxonomy may have been distorted by unrepresentativeness in original selection of terms. Thus our confidence in the present taxonomy is greatest at the level of the four main categories and diminishes as we traverse down the tree structure.

The Basic Interface Model

The first branch reflects the view that the user interface is composed of the seven fundamental components:

1.1 Actions
1.2 Behaviours
1.3 Contexts
1.4 Displays
1.5 Effects
1.6 Forms
1.7 Goals

Each of these fundamental components may then be further expanded into subcomponents. The actions of the computer application may be broken down into traditional components of processing, i.e., cpu, I/O, and peripherals, leading to the following categorization:

1.1 Actions
  1.1.1 Computation
  1.1.2 Storage
  1.1.3 Retrieval
  1.1.4 Operation of Peripheral Devices

The behaviors of the user are a little bit more difficult to characterize. In this classification, we have chosen to emphasize tradition concerns in HCI, i.e, the problems of navigation, information seeking, interaction styles, and input devices. Of these four problems, the issue of user navigation is probably the least understood. We will provisionally characterize user navigation as the way in which the user moves around the conceptual structure of the interface. In practice, there will be a high degree of overlap between user navigation and interaction styles, since the intentions of the user during navigation will have to be communicated to the interface via some style of interaction.

1.2 Behaviours
  1.2.1 User Navigation
    1.2.1.1 Selection from Map or Display
    1.2.1.2 Move to more general command or concept
    1.2.1.3 Move to more detailed command or concept
    1.2.1.4 Move to associated concept or command
    1.2.1.5 Switch between modes, windows or environments
    1.2.1.6 Ask for Help
    1.2.1.7 Move through a sequence of commands or information
  1.2.2 Interaction Styles
    1.2.2.1 Direct Manipulation
      1.2.2.1.1 Icons
      1.2.2.1.2 Engagement
      1.2.2.1.3 S-R Compatibility
    1.2.2.2 Command Driven
      1.2.2.2.1 Command Languages
        1.2.2.2.1.1 Command Syntax
        1.2.2.2.1.2 Language Semantics and Expressiveness
        1.2.2.2.1.3 Shortcuts and Macros
      1.2.2.2.2 Command Menus
    1.2.2.3 Form Filling
      1.2.2.3.1 Query by Example
    1.2.2.4 Menus
      1.2.2.4.1 Menu Design
      1.2.2.4.2 Menu Maps
      1.2.2.4.3 Menu Selection
    1.2.2.5 Conversational
    1.2.2.6 Graphical Structures
  1.2.3 Input Devices
    1.2.3.1 Data Entry
      1.2.3.1.1 Keyboard
      1.2.3.1.2 Function Keys

1.2.3.1.3 Bar Codes
1.2.3.1.4 Digitizing Tablet
1.2.3.2 Pointing Devices
1.2.3.2.1 Direct Pointing
1.2.3.2.1.1 Lightpen
1.2.3.2.1.2 Touch Screen
1.2.3.2.2 Indirect Pointing
1.2.3.2.2.1 Mouse
1.2.3.2.2.2 Trackball
1.2.3.2.2.3 Joystick
1.2.3.2.2.4 Graphics Tablet
1.2.3.3 Speech Recognition

The next category (contexts) is an aspect of the basic interface model that has not received much attention in HCI research. What are the types of contexts that user behaviors can occur in? It turns out that there are many different ways to classify the kinds of contexts that occur in HCI. We will list a few that seem to have a strong effect on performance. These include; task complexity, temporal constraints, hardware or software malfunctions, knowledge of results (i.e., are the task relevant effects of previous behavior visible to the user?)

1.3 Contexts
1.3.1 Task Complexity
1.3.2 Temporal Constraints
1.3.3 System Malfunctions, Limitations, and Capabilities
1.3.4 Knowledge of Results

The category of displays deals with a new and relatively unknown area. While the general problem of displays has been around for a long time, recent technological advances have made it possible to combine information from a number of different media and to compose it so that the screen and sound spectrum can become a collage of information from different sources. Provisionally, we have divided the category of displays into the topics of multimedia, screen design, and display composition, recognizing that there will be some overlapping issues between these subcategories.

1.4 Displays
1.4.1 Multimedia
1.4.1.1 Auditory Displays
1.4.1.2 Video
1.4.1.3 2-D Graphics
1.4.2.4 3-D Graphics
1.4.2.5 Text Display
1.4.2 Screen Design
1.4.2.1 Screen Complexity
1.4.2.2 Window Design
1.4.2.2.1 Sunview
1.4.2.2.2 XWindows
1.4.2.2.3 News
1.4.2.2.4 Andrew
1.4.3 Display Composition
1.4.3.1 Prioritization of Dispayable Information
1.4.3.2 Sequencing and Positioning of Information Sources
1.4.3.3 Orientation of Attention to Displays
1.4.4 Sensory Coding of Information and Visual Cues
1.4.4.1 Icons
1.4.4.1.1 Icon Design
1.4.4.1.2 Icon Libraries
1.4.4.1.3 Moving Icons (Micons)
1.4.4.2 Earcons

The distinction between displays and effects reflects the difference between information presented for its own sake and feedback about the operations of the application in response to user behaviors. Many of the details about effects will be application specific, so our classification will list only those types of effect that generally occur across applications.

1.5 Effects
1.5.1 Response Time
1.5.2 Errors
1.5.2.1 Error Messages
1.5.2.2 Error Prevention

As stated earlier, forms are the models in which actions, effects, and displays are embedded. The role of visual metaphors and spatial models as types of form seems obvious. However, visualizing information has also been included as a form because it deals with the task of making the user understand the structure and gross features of information rather than the details of the information. This is consistent with the general role of forms which is to assist the user in forming an accurate conceptual model of how the interface, and the information and functions that it contains, is structured. Examples of the work on visualizing information include use of automatic icons (Fairchild, Meredith, and Wexelblat, 1988a) and artificial realities (Fairchild, Meredith, and Wexelblat, 1988b).

1.6 Forms
1.6.1 Non-Spatial Metaphors
1.6.1.1 Books
1.6.1.2 Cards
1.6.1.3 Files
1.6.1.4 People and Situations
1.6.1.5 Desktops
1.6.2 Spatial Models
1.6.2.1 Maps
1.6.2.2 Rooms
1.6.2.3 Freeways
1.6.2.4 Three-Dimensional Spaces
1.6.3 Visualizing Information
1.6.3.1 Automatic Icons
1.6.3.2 Simulations and Artificial Realities

Goals represent the motivating forces behind HCI. Task analysis is the process by which a task is decomposed into sequences of goals and subgoals. It is typically a normative process based on a rational model of the task. In contrast, goal identification is a descriptive process that infers goals on the basis of user behavior while performing the task. Methods for goal identification include verbal protocol analysis, interviewing, error and critical incident analysis, and response time studies, where goals are inferred on the basis of preparatory pauses in transaction logs (e.g., Eberts, 1987).

1.7 Goals
1.7.1 Normative Task Analysis
1.7.2 Descriptive Goal Identification

1.7.2.1 Verbal Protocol Analysis
1.7.2.2 Interviewing
1.7.2.3 Error and Critical Incident Analysis
1.7.2.4 Response Time Studies

Cognitive Engineering

The second branch of the tree is cognitive engineering. This refers to the process of applying the models and findings of cognitive science to the task of analyzing and designing user interfaces. The cognitive engineering branch is broken down into three sub-branches:

2.1 Cognitive Science
2.2 Normative Models
2.3 Descriptive Models

The three sub-branches of cognitive engineering may then be further broken down into subtopics as shown below:

2.1 Cognitive Science
2.1.1 Cognitive and Experimental Psychology
2.1.1.1 Experimental Design and Analysis
2.1.1.2 Human Intelligence and Abilities
2.1.1.3 Personality and Motivation
2.1.1.4 Human Information Processing
2.1.1.4.1 Learning
2.1.1.4.2 Memory
2.1.1.4.3 Decision Making
2.1.1.4.4 Problem Solving
2.1.1.4.5 Attention
2.1.1.4.5.1 Selective Attention
2.1.1.4.5.2 Divided Attention
2.1.1.4.5.3 Focused Attention
2.1.1.4.5.4 Attentional Resources
2.1.1.5 Perception
2.1.2 Artificial Intelligence
2.1.2.1 Knowledge Representation
2.1.2.1.1 Semantic Nets
2.1.2.1.2 Frames
2.1.2.1.3 Production Rules
2.1.2.1.4 Scripts
2.1.2.2 Symbolic Programming
2.1.2.2.1 Lisp
2.1.2.2.2 Prolog
2.1.2.2.3 SmallTalk
2.1.2.3 Knowledge Engineering
2.1.2.3.1 Knowledge Acquisition
2.1.2.3.2 Machine Learning
2.1.2.3.3 Inference
2.1.2.4 Machine Vision
2.1.3 Language Understanding
2.1.3.1 Syntax
2.1.3.2 Semantics
2.1.3.3 Discourse Analysis
2.1.3.4 Text Analysis
2.1.3.5 Language Acquisition
2.1.4 Neuroscience
2.1.4.1 Neurophysiology
2.1.4.2 Action
2.1.4.3 Human Vision
2.1.4.4 Learning and Memory
2.1.4.5 Hemispheric Differences
2.1.5 Philosophy
2.1.5.1 Ontology
2.1.5.2 Epistemology

At present, the section of normative models simply lists some of the more prominent approaches. Eventually, it would be desirable to classify these approaches into different types of model, e.g., high-level vs. low-level models.

2.2 Normative Models
2.2.1 GOMS
2.2.2 Model Human Processor
2.2.3 Keystroke-level Model
2.2.4 ACT*
2.2.5 Task Action Grammar

Descriptive models of HCI have received a lot of attention in recent years. Basic distinctions can be made between the user's mental model of the system, the conceptual model of the system as seen by the designer, and the user model that the system has in interpreting user inputs and handling the interaction. Research methods for identifying these descriptive models tend to be the same as for those used in goal identification (1.7.2 above).

The subcategory of knowledge in this section of the taxonomy refers to methods for handling descriptive knowledge that is relevant in an HCI application. Knowledge acquisition and knowledge engineering are terms that have been used with reference to building expert systems, but they can also be applied to processes for discovering and using knowledge within HCI. Knowledge compilation is the process of streamlining and automating knowledge for particular tasks (Anderson, 1983). In terms of HCI, it can be used to describe both the compilation of the users knowledge about the application and the interface and the system's knowledge about the user and the task.

2.3 Descriptive Models
2.3.1 Mental Model
2.3.2 Conceptual Model
2.3.3 User Model
2.3.3.1 Usage Profiles
2.3.3.2 Resources and Workload
2.3.3.3 Expertise
2.3.4 Domain Model
2.3.4.1 Knowledge Acquisition
2.3.4.2 Knowledge Engineering
2.3.4.3 Knowledge Compilation

Interface Engineering

The next branch of the tree to be dealt with is interface engineering. This is the portion of the user interface research taxonomy which deals with guidelines and approaches for interface design (engineering). The five branches of interface engineering are:

3.1 General Guidelines
3.2 Interface Structuring
3.3 Interface Training
3.4 Interface Evaluation
3.5 User Interface Development Systems and Tools
3.6 Interface Engineering Techniques

The general guidelines represent a conceptual framework that guides HCI research and design. Some of these guidelines are based on empirical findings, others represent a consensus view of the HCI research community. Some of the guidelines have even reached the stage of being axiomatic. For instance, structuring information into manageable chunks (e.g., menu selections) and hiding unnecessary information from the user are close to being axioms of HCI.

The general guidelines shown below should be recognizable to most HCI researchers. The Do's and Don'ts refer to various lists of guidelines that have been published (e.g., Smith and Mosier, 1986). Computer anthropomorphism refers to the problem of making the computer seem more like a person than it really is, which can eventually lead to failed expectations and frustration on the part of the user.

3.1 General Guidelines
- 3.1.1 Do's and Don'ts
- 3.1.2 Excess Functionality
- 3.1.3 Cognitive Compatibility
- 3.1.4 Computer Anthropomorphism
- 3.1.5 User-Centered Design

Interface structuring is a category that is somewhat related to the issue of user navigation (1.2.1 above). Possibilities for user navigation generally depend on the structuring principles and types of conceptual structuring that are available in the interface. Dialogue design is an aspect of conceptual structuring concerned with how users make choices, enter data and the like. The results of a dialogue will often be a transition from one state or situation to another.

The dominant structuring principles at present in HCI are hierarchies (e.g., menu systems), linear sequences, and networks or hypermedia. Look up or relational tables may also be used for structuring. Under the others subcategory might be included more specialized structuring methods such as spreadsheets.

3.2 Interface Structuring
- 3.2.1 Conceptual Structuring
  - 3.2.1.1 Dialogue Design
  - 3.2.1.2 States or Situations
- 3.2.2 Structuring Principles
  - 3.2.2.1 Hierarchies
  - 3.2.2.2 Hypermedia
  - 3.2.2.3 Linear Sequences
  - 3.2.2.4 Tables
  - 3.2.2.5 Others

The usability of an interface will be determined by the combination of its design, the task complexity, the user skills, and the availability of interface training, among other things. Interface training is a broad category, but relevant aspects include online tutorials, help facilities, and documentation.

3.3 Interface Training
- 3.3.1 Online Tutorials
- 3.3.2 Help Facilities
- 3.3.3 Documentation

Interface evaluation is one of the most vexing issues in HCI. In general one can distinguish between formative evaluation, where one is seeking to improve the design, and summative evaluation, where one is trying to assign an overall figure of merit to a developed application.

Face Validity refers to whether or not the proposed interface design has an acceptable look and feel (particularly to potential users). An adequate first impression of the proposed interface can often be conveyed with storyboards and similar techniques. Wizard of Oz experiments are used to construct a simulation of how the interface will behave prior to detailed programming. An example of a Wizard of Oz experiment would be a situation where the application is simulated by having an experimenter type in input to the user's terminal in accordance with the design. Prototypes may also be used to evaluate proposed designs. Rapid construction of prototypes is often possible using the methods listed in section 3.5 below.

For summative analysis of the interface once it has been constructed, the preferred method is performance analysis. Yet it may sometimes be difficult to specify an adequate benchmark task that provides an adequate test of the usability of the interface. It helps if there are alternative implementations of the application to compare the current interface with, since it is otherwise difficult to separate the effects of the interface from the effects of task complexity and the design of the application.

As a result of difficulties in performance analysis, questionnaires are probably the most frequently used method of summative evaluation of user interfaces. However, questionnaires provide a subjective evaluation of interfaces which is often greatly influenced by the type of questions asked and the way in which the questions are phrased.

One issue in summative evaluation is "when to evaluate?" An interface that the user initially finds difficult to use may eventually prove to be easy to use and have high functionality for the user. Learning curve analysis looks at how use of the interface improves over time. Thus instead of analyzing performance or administering questionnaires at a single point in time, several evaluations may be made over a period of time as the user becomes more familiar with the system. Thus learning curve analysis takes the other methods of summative evaluation and applies them over a time period to get a more complete profile on how usability changes with usage. In most situations the usability of an interface on the first day of use will not be as important as the slope of the learning curve over subsequent days and the amount of proficiency attained by users after extended use.

3.4 Interface Evaluation
- 3.4.1 Formative Evaluation
  - 3.4.1.1 Face Validity
  - 3.4.1.2 Wizard of Oz experiments
  - 3.4.1.3 Prototype Evaluation
- 3.4.2 Summative Evaluation
  - 3.4.2.1 Performance Analysis
  - 3.4.2.2 Learning Curve Analysis
  - 3.4.2.3 Questionnaires and Interviews

In the past, user interfaces have been notoriously difficult to construct. Software developers have often found that the majority of time spent in implementing an application is actually spent on the interface. Thus there has been a great deal of interest in the development of systems and tools that can both speed up the development of user interfaces and increase their level of usability. The taxonomy currently divides these systems and tools into three classes; user interface toolkits, user interface management systems (UIMSs), and Specification Techniques.

3.5 User Interface Delevelopment Systems and Tools
- 3.5.1 User Interface Toolkits
  - 3.5.1.1 HP Toolkit
  - 3.5.1.2 Macintosh Toolbox
  - 3.5.1.3 Others
- 3.5.2 User Interface Management Systems
  - 3.5.2.1 Presentation Management
  - 3.5.2.2 Behavior Management
  - 3.5.2.3 Automated Layout
- 3.5.3 Specification Techniques
  - 3.5.2.1 Grammar-based Formalisms
    - 3.5.2.1.1 Command Language Grammar
    - 3.5.2.1.2 Backus-Naur Formalism
  - 3.5.2.2 Network Formalisms
  - 3.5.4.3 State Transition Networks
  - 3.5.4.4 Petri Nets
  - 3.5.2.5 Graphical Specification

The final sub-branch of interface engineering refers to techniques that can enhance the interface engineering process, these include iterative design and rapid prototyping.

3.6 Interface Engineering Techniques
- 3.6.1 Iterative Design
- 3.6.2 Rapid Prototyping

Human-Computer Interaction Applications

At present there is no standard user interface that can be used across all applications. The type of application has a major impact on the way that the user interface is constructed. For instance, real-time applications have to utilize methods for getting critical time-dependent information to the user or operator. Thus at a detailed level, much of the information and techniques in HCI will be application-specific, necessitating the incorporation of applications within the hierarchy. Areas that have received attention from researchers in human-computer interaction include:

4.1 Real Time Applications
4.2 Information Technology
4.3 Advanced Programming
4.4 Manufacturing and Industry
4.5 Computer Assisted Learning

Further classification is shown below, however it is recognized that this portion of the taxonomy is relatively incomplete.

4.1 Real Time Applications
- 4.1.1 Process Control
- 4.1.2 Satellite Monitoring
- 4.1.3 Video Games
- 4.1.4 Military Applications

4.2 Information Technology
- 4.2.1 Management Information Systems
- 4.2.2 Decision Support Systems
- 4.2.3 Database Management Systems
  - 4.2.3.1 Relational Databases
  - 4.2.3.2 Object-Oriented Databases
- 4.2.4 Expert Systems and Intelligent Databases
  - 4.2.4.3 Data Analysis
  - 4.2.4.4 Consultation
- 4.2.5 Information Retrieval
  - 4.2.5.1 Querying
  - 4.2.5.2 Browsing
- 4.2.6 Learning Support Environments
- 4.2.7 Text Editing
- 4.2.8 Office Information Systems
- 4.2.9 Simulation
  - 4.2.9.1 Discrete-Event Simulation
  - 4.2.9.2 Continuous Systems Simulation

In the classification of information technology above we have outlined four subcategories of information seeking. We include data analysis as information seeking here because it is a method of extracting and obtaining information. It may involve a number of summarization methods including statistics, operations research, time series analysis, and graphical presentations. Consultation refers to the process of getting information from an advisory source such as an expert system or a decision support system.

4.3 Advanced Programming
- 4.3.1 Visual Programming
- 4.3.2 Knowledge Engineering and Logic Programming
- 4.3.3 Computer-Aided Software Engineering (CASE)
- 4.3.4 Simulation

4.4 Manufacturing and Industry
- 4.4.1 Computer-Aided Design (CAD)
- 4.4.2 Computer-Integrated Manufacturing (CIM)
- 4.4.3 Process Planning
- 4.4.4 Inventory Management
- 4.4.5 Transportation and Shipping
- 4.4.6 Accounting and Billing
- 4.4.7 Payroll Systems

4.5 Computer Assisted Learning
- 4.5.1 CAI (computer-assisted instruction)
- 4.5.2 Intelligent Tutoring Systems and Intelligent CAI
- 4.5.3 Learning Support Environments

Conclusions

This paper represents a first attempt to formulate a taxonomy of human-computer interaction. This taxonomy is offered as a provisional framework that should be revised according to the consensus views of the human-computer interaction research community. After much refinement, this taxonomy may ultimately be of considerable assistance in standardizing terminology, clarifying disputes based on terminological differences, and providing a basis for research and instruction in this important but complex area.

Acknowledgments

The author would like to thank the user interface group,

and Gurminder Singh and John Waterworth in particular, at the Institute of Systems Science for their comments on earlier versions of this taxonomy. This article was prepared while the author was a visiting scientist at the Institute of Systems Science, National University of Singapore. Comments on this taxonomy would be gratefully appreciated and will be taken into consideration in future revisions.

References

Anderson, J.R. (1983) **The Architecture of Cognition.** Cambridge, Mass.: Harvard University Press.

Baecker, R.M. and Buxton, W.A.S. (Eds.) (1987) **Readings in Human-Computer Interaction: A Multidisciplinary Approach.** Los Altos, CA: Morgan Kaufmann.

Bailey, R.W. (1989) **Human Performance Engineering.** 2nd Edn. Englewood Cliffs, N.J.: Prentice Hall.

Carroll, J.M. (Ed.) (1987) **Interfacing Thought: Cognitive Aspects of Human-Computer Interaction.** Cambridge, Mass.: MIT Press.

Carter, J.A. Jr. (1986) *A taxonomy of user-oriented functions.* **International Journal of Man-Machine Studies,** 24, 195-292.

Fairchild, K.M., Meredith, L.G., and Wexelblat, A.D. (1988a) **A formal structure for automatic icons.** MCC Technical Report Number STP-311-88, September 1988.

Fairchild, K.M., Meredith, L.G., and Wexelblat, A.D. (1988b) **The tourist artificial reality.** MCC Technical Report Number STP-310-88, November 1988.

Hartson, H.R. and Hix, D. (1989) *Human-computer interface development: Concepts and systems for its management.* **ACM Computing Surveys,** 21, 5-91.

Helander, M. (Ed.) (1988) **Handbook of Human-Computer Interaction.** Amsterdam: North-Holland.

Norman, D.A. and Draper, S.W. (Eds.) (1986) **User-Centered System Design.** Hillsdale, NJ: Erlbaum.

Shneiderman, B. (1987) **Designing the User Interface Strategies for Effective Human-Computer Interaction.** Reading, Mass.: Addison-Wesley.

Smith, S.L. and Mosier, J.N. (1986) **Guidelines for designing user interface software.** Tech. Rep. ESD-TR-86-278. Hanscom Air Force Base, Mass.: USAF Electronic Systems Division (NTIS No. AD A177198)

Stillings, N.A., Feinstein, M.H., Garfield, J.L., Rissland, E.L., Rosenbaum, D.A., Weisler, S.E., and Baker-Ward, L. (1987) **Cognitive Science: An Introduction.** Cambridge, Mass.: MIT Press.