University of Alberta

Human-Computer Pragmatics:
From Habermas's Theory to User Centric Design

by

Karl Leif Anvik ©

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Arts
in
Modern Languages and Cultural Studies

Humanities Computing

Edmonton, Alberta
Fall 2007

Your file Votre référence
ISBN: 978-0-494-33107-1
Our file Notre référence
ISBN: 978-0-494-33107-1

Abstract

This thesis proposes a theory of human-computer pragmatics based on Habermas's theory of universal pragmatics. Human-computer pragmatics views the user and the designer as participants in a conversation within a social sphere. As designers avoid strategic action and conform to the validity claims of comprehensibility, truth, truthfulness and normative right, they are more likely to communicate the information necessary for successful operation of the system.

A practical application of the proposed theory is presented through an evaluation of a prototype interface for a multilingual information retrieval system. The evaluation of the prototype interface suggests that constant window scrolling may reduce comprehensibility; the use of controlled vocabularies in this system improves its claim to truth; the application of matrices improves its claims to truthfulness; and finally, improving the order that interface elements are displayed to a left-to-right, top-to-bottom reading sequence based on task improves the claim to normative right.

Acknowledgements

Firstly, I would like to express my appreciation for the work of my advisor, Stan Ruecker, who inspired and guided much of this work. His help was invaluable at every phase of its development, particularly through his ideas on browsing interfaces and the organization of this thesis. His many helpful thoughts and insight were important as this work progressed.

In addition, this work could not have been completed without those that helped in the development of the multilingual information retrieval system prototype interface. Ali Shiri and Ximena Rossello deserve recognition in this regard.

I would also like to thank the members of my committee for taking time out of their busy schedules to review this work and provide additional comments and suggestions.

Finally, I must acknowledge my wife Cassandra, whose love, support and patience helped see this work through to its completion. While unfamiliar with much of the material found within its pages, she read every word and offered many helpful comments along the way. My daughter Clara should also be acknowledged for providing many much needed interruptions and also for her patience when daddy was too busy to play.

Contents

List of Figures

Chapter 1 : Introduction

Over the course of approximately five millennia, humans have generated and recorded a plenitude of information. The earliest known repository of written records dates back to the Sumerian civilization located in southern Mesopotamia over five thousand years ago. Personal libraries first appeared in classical Greece around the fifth century BCE. Around 280 BCE, the Library of Alexandria in ancient Egypt is thought to have had over 750,000 documents (*History magazine* 2001). In modern times, technology has made it possible to navigate a myriad of documents very quickly. In 1945, Vannevar Bush published *As we may think*, where he illustrated his vision of a sophisticated device called a *memex*, "an enlarged intimate supplement to [an individual's] memory", where a person could store "all his books, records, and communications ... with exceeding speed and flexibility" (Bush 1945). Today the "memex" has materialized in the form of a personal desktop computer. The Internet has emerged as an absolutely colossal store of information. A computer connected to the Internet provides its user with the ability to access an enormous reservoir of knowledge. If one were to enter the word "albatross" into Google's search engine, it will find approximately 4,090,000 documents in less than one tenth of a second.

Over the past century, advances in air travel and communication technologies such as the radio, telephone, satellite transmission, and most recently the Internet, have brought about an extraordinary evolution of our society. People are interacting more and more with others from around the globe, transforming our world into what is often referred to as "the global village". With the growth of the Internet, communication with others is becoming even easier. Equally significant is the ability for individuals to access recorded information from around the globe. With a few strokes of the keyboard, someone in Edmonton, Canada can read a newspaper from Brazil, peruse an Italian travel magazine, or study political commentary in a Russian weblog.

As technological improvements provide more refined methods for archiving and publishing information on a massive scale, the need for developing sophisticated and effective tools for searching this information has increased substantially. The Internet is a source of abundant information and serves as a vast repository for text, audio, images and video on a wide range of topics. Due to the proliferation of this kind of information, libraries have leveraged technologies for archiving digital media and now provide access to a host of databases that contain citations and digitisations of academic periodicals, conference proceedings, doctoral dissertations, musical scores, sound recordings and other archived materials.

The term Information Retrieval (IR) has been around since the early 1950s. It is not a coincidence that this time period is also considered to be the beginning days of the computer era. The computer has forever changed the way we organize and archive information. According to Chowdhury (1999), an IR system can be defined as a system that "stores and retrieves information," where the system refrains from informing the user specifically on a particular subject, but merely informs the individual of the existence and location of documents related to a specific request. Over the centuries the sum of knowledge produced and recorded by humankind has increased to an incomprehensible level. Over the past several decades, information scientists have been developing systems that employ a wide range of models and techniques to aid in the management and retrieval of massive amounts of information. While it is now more convenient to access many different kinds of materials, architectural improvements of IR systems have not always been accompanied by advancements in usability of such systems.

In early IR system design, the role of the end-user was minimized, giving way primarily to architectural and algorithmic considerations. Although work in this area has produced systems that are better at retrieving relevant information, the user experience was not a focus of improvement. However, over the past number of years the importance of user-centric design has become more prominent and the usability issue has evolved in various research communities (Sjöström and Ågerfalk 2004). One such research community where the focus has yet to show significant progress with respect to user interface design strategies is the area of Multilingual Information Retrieval (MLIR) – an area of research where the main concern is to explore methods that facilitate retrieval of documents that have been written in different languages.

As our society evolves into a global community it is becoming more important to be able to publish and access information on a global level. Despite many technological improvements, language barriers are still making it difficult to bring research communities together where they are more capable of sharing ideas and information. For example the Internet has been primarily responsible for the recent growth in sharing information across both political and cultural boundaries. It has minimized many obstacles encountered before its creation such as differences in network and computer systems. As computing standards gain importance, these kinds of differences are becoming less problematic. At the same time, despite these technological advances, language barriers between many cultures continue to make it difficult to share information on a global level. In order to facilitate information sharing despite different forms of communication, information systems must be able to cross language boundaries. Users must be able to obtain information recorded in multiple languages by entering queries in one language then expect the system to return documents relevant to their query regardless of language.

User interface design of IR systems introduces many challenges beyond those of the organization of information, the architectural design of the computer system and algorithmic considerations. The difference between human-computer communication and computer-computer communication lies in the fact that humans communicate much differently than computers. Computer systems communicate with one another in a precise manner, relying on logic, set protocols and well defined standards in order to execute commands, invoke procedures and transmit information. Humans, on the other hand, communicate in a much more complex fashion than computer systems, and it is difficult to measure and characterize their motivations and behaviours (Hearst 1999). In order to facilitate communication between human beings and computers, some kind of intermediary interface must be present. This interface must be able to deal with the extremely structured form of communication of computers as well as the imprecise form of communication of human beings. The system designer then becomes, in a sense, a translator – translating between the communication structures of human beings and those of computer systems.

Technology is becoming ubiquitous in our society and more people who would previously have been considered non-traditional users of technology are finding it necessary to use and interact with computers. As technology continues to advance and becomes more feature-rich, the usability problem becomes increasingly complex. This is one of many driving forces for the increasing attention given by designers to the usability of information systems (Rimmer 2004). No matter what the application or device, whether it be a word processor, a cell phone, or a portable digital music player, designers are becoming more aware of the importance of making these technologically complex devices simple enough for the average user. This understanding has led to the emergence of an entirely new field of study called Human-Computer Interaction (HCI). The primary goal of HCI is to enhance the interaction between humans and computers by making computer systems more user-friendly. Researchers in the area of HCI are concerned with developing methodologies and processes for designing, implementing and evaluating user interfaces.

When it comes to the importance of usability, IR systems are no exception. In order for an IR system to maximize usability (and therefore maximize efficiency and effectiveness), the system must instil confidence in the user. As Shneiderman and Plaisant explain (1997):

> Well designed, effective computer systems generate positive feelings of success, competence, mastery, and clarity in the user community. When an interactive system is well-designed, the interface almost disappears, enabling users to concentrate on their work, exploration, or pleasure.

This kind of user interface is what designers should work towards. An interface that "almost disappears" is unquestionably the holy grail of human-computer interface design.

As the concepts of Human-Computer Interaction and usability have increased in importance, the research dedicated to this problem has evolved and matured. Specialists in this field have adopted and incorporated theories and research from a variety of disciplines, including psychology, biology, semiotics and linguistics. One theoretical framework that has rarely been drawn from to enhance HCI theory is universal pragmatics, in particular as laid out by Jürgan Habermas with his conception of the ideal conversation. The primary focus of this thesis will be to explore how Habermas's ideas can influence the design of user interfaces of Information Systems, in particular Multilingual Information Retrieval systems.

The purpose in the approach of applying Habermas's theory to user-interface design is that interaction with a computer can be viewed as a conversation. When an individual interacts with a piece of technology, the person enters into a conversation with that piece of technology. This conversation with the device or system is facilitated by the designers of the system. Both the system and the interface designer transmit certain messages to the user through a number of different design decisions. Just as in a traditional verbal conversation, these decisions may introduce various forms of interference into the communication channel, resulting in a conversation where the user has difficulty understanding what is being said. If the system does not communicate its knowledge clearly and effectively, the user may feel untrusting of the system.

The user not only feels unsuccessful, but possibly incompetent, as though the lack of success is somehow his or her fault. In actuality, the fault may not be a result of the user's incompetence, but is a direct result of a poorly designed interface – one that does not communicate effectively to the user on how to interact with the system. With a better understanding of how communication theory applies to user interface design, designers can create interfaces that are more effective at communicating with the user, decreasing levels of frustration, instilling greater confidence in the user's abilities and ultimately achieving better results and productivity.

The contents of this work will bring together theories and research from three main research areas: multilingual information retrieval, human-computer interaction, and communication action theory. Chapter 2 will provide some background information and give an outline of what research has been done in each of these areas. It will look at the use of natural language processing, machine translation and multilingual thesauri in the area of multilingual information retrieval. With respect to HCI, the theoretical background and motivations surrounding metaphors, affordances and phenomenology will be explored. Finally, some background knowledge of speech act theory, universal pragmatics and current research applying these principles to HCI will be given.

The contents of Chapter 3 will offer some of my own ideas about how to apply Habermas's theory of universal pragmatics to user-interface design to achieve the 'ideal conversation' between user and system. In chapter 4, a MLIR interfaces will be examined to see how well it conforms to the principles explored in the previous chapter. Suggestions on how to improve each interface in order to support the ideal conversation will also be given. Finally, concluding remarks and potential future work will be discussed in chapter 5.

Chapter 2 : Background and Previous Research

Before examining the intricacies of universal pragmatics and its application in the area of human-computer interaction, it is necessary to provide some background information on this research domain. The specific example of Multilingual Information Retrieval (MLIR) user interfaces to be examined in chapter 4 will necessitate the exploration of the general aspects of this field. This chapter will outline previous work that has taken place in the areas of human-computer interaction and MLIR, including a number of key theories that have significantly influenced these areas of research. It will begin with an introduction to the field of MLIR, some of its challenges, previous research and an outline of current approaches, necessitating a brief introduction to machine translation. It will then move on to an outline of contributions to human-computer interaction by individuals prominent in this field such as Gibson, Erickson, Nielson, Norman, Shneiderman, Winograd and Flores. This chapter will then conclude by visiting the basic principles of speech act theory, Habermas's universal pragmatics and finally its application to the field of information technology and human-computer interaction.

2.1 – Multilingual Information Retrieval

As stated earlier, the real world example of MLIR will be used to explore how the marriage of universal pragmatics and human-computer interaction principles helps designers communicate to users, ultimately producing more effective user interfaces. Before delving into the specifics of MLIR user interfaces, it will be important to understand a few basic principles of the field. MLIR is essentially a convergence of two fields, namely information retrieval and machine translation.

As a research domain, MLIR (sometimes referred to as Cross Language Information Retrieval, or CLIR) involves researchers from a variety of disciplines including library and information sciences, computer sciences and linguistics. While there is currently no commonly accepted definition for MLIR, notable contributors to the field, Hull and Grefenstette (1996) have compiled five different scenarios that they consider fall under MLIR:

> (1) [Information retrieval] in any language other than English.
>
> (2) [Information retrieval] on a parallel document collection or on a multilingual document collection where the search space is restricted to [a controlled vocabulary in] the query language.
>
> (3) [Information retrieval] on a monolingual document collection which can be queried in multiple languages.
>
> (4) [Information retrieval] on a multilingual document collection, where queries can retrieve documents in multiple languages.
>
> (5) [Information retrieval] on multilingual documents, i.e. more than one language can be present in the individual documents.

The first scenario (1) is the least complex of the five outlined by Hull and Grefenstette, and its appearance in this list is debatable. One might question how information retrieval in a language other than English differs from information retrieval in English. Can information retrieval principles not be applied to information retrieval in other languages? One could argue that the fundamental difference between information retrieval and MLIR is that a barrier existing between two languages must be resolved. In this case it does not seem as though there is a language barrier as the language of the search terms provided by the user is equivalent to the language of the documents being searched.

In each of the remaining cases above (2-5), there must be some means of overcoming a language barrier. The language barriers in each of the above definitions can be described as follows: (2) The user is searching a repository of documents, but the repository (the collection) contains documents in two or more languages, and the user is presented with a predefined set of valid search terms. These search terms must somehow map to equivalent terms in each language supported by the system. In this situation search terms are moving from one language to potentially many languages (one-to-many). (3) The repository contains documents in one language only, but the system will accept search terms in more than one language. Search terms entered by the user must be translated by the system into the target language as defined by the documents in the repository. Here, the system is moving from potentially many languages to one target language (many-to-one). (4) This situation is similar to that of (2), but in this case the search terms are not restricted to any predefined set of terms. The user can enter any search term in any language supported by the system. Again the system is moving from one language to potentially many languages (one-to-many). (5) This is the most complex situation of the set. As in (4), the user can enter a search term in any language supported by the system. Because the repository contains documents in many languages, and each document can potentially contain text in more than one language, the system must be able to translate search terms both to and from a variety of languages (many-to-many). Chapter 4 will consider a user interface for a MLIR system that corresponds with definition (2) and will explore techniques for developing an effective user interface for such a system.

As stated previously, MLIR is a convergence of two distinct research areas: machine translation and information retrieval. There has been much commentary and skepticism on the ability of computers to be able to fully automate high quality translations of the human language. As Raley (2003) so eloquently explains, “No reasonable person thinks that a machine translation can ever achieve elegance and style. Pushkin need not shudder.” Literary works can be considered the most complex type of text to translate, and it is reasonable to say that machine translation systems are a very long way from being able to fully automate translations of these kinds of texts.

While computers may never be able to translate literary works in a completely automatic process, these machines have some success working with less complicated texts. For example, scientific texts might be considered less complex than literary works. When Warren Weaver proposed the notion of a computer that could translate languages in 1947, he addressed this comparison between scientific texts and their more complicated literary counterparts. Weaver stated, “Even if it would translate only scientific material (where the semantic difficulties are very notably less), and even if it did produce an inelegant (but intelligible) result, it would seem to me worth while” (Raley 2003).

While MLIR is closely related to machine translation, it is a much easier problem to solve. Grefenstette (1998) explains why this is so:

> They [MT and MLIR] have in common that systems developed with either approach in mind must produce versions of the same text in different

> languages but machine translation systems must respect two additional constraints of choosing one and only one way of expressing a concept, and of producing a syntactically correct version of the target language text that reads like naturally created text. A Cross Language Information Retrieval system has any easier job, needing only produce the translated terms to be fed to an information retrieval system, with little worry about presentation of its intermediate results for human consumption.

Grefenstette points out that the most difficult obstacle to overcome in machine translation is that human language is difficult for computers to understand and process. Fortunately MLIR systems are required to manage what may be considered the simplest aspect of machine translation.

In most information retrieval systems, queries are typically formed as a sequence of terms, not adhering to a grammatical structure of any kind. For example, "Show me all the essays that contain commentary on existentialism in Dostoevsky's novel *The idiot,*" is not how one would formulate a query in most contemporary information retrieval systems. While it is true that researchers are trying to leverage some advancements in the field of Natural Language Processing to develop information retrieval systems that could accept such queries, most current information retrieval systems do not perform at this level of complexity. A more appropriate query might simply look like this: "Existentialism Dostoevsky 'The Idiot'", or if Boolean operators are used, "Existentialism and Dostoevsky and The Idiot". There may be additional options that allow the user to restrict searches to just journal articles and exclude books or dissertations. This type of query is much less complicated; a MLIR system would need only to translate three terms, not paying any heed to grammatical correctness.

While the translation aspect of a MLIR system is typically less complicated than a fully automated MT system, it is still not without its own set of challenges. Grefenstette (1998) outlines the three fundamental problems of MLIR as follows:

(A) Understanding how terms are expressed in different languages.

(B) Deciding which problems should be retained.

(C) Properly weighting the importance of translation alternatives.

While some of these challenges are unique to MLIR systems, others are similar to those found in complementary domains. For example, MLIR must address the issue of resolving ambiguities in language, an issue that must be addressed by machine translation systems as well. Some examples of ambiguous phrases that a MT system might encounter could be (Arnold 2003):

(1) 'Sam put the sheep in the pen'.

(2) 'The minister stated that the proposal was rejected yesterday'.

(3) 'The police refused to let the women demonstrate because they advocated violence'.

Each of these phrases requires certain judgements to be made in order to understand what is really meant by the speaker/author. In example (1), is the pen in question an animal enclosure or a writing utensil? In example (2) was the minister making a statement yesterday, or was the proposal rejected yesterday? In example (3), was it the police advocating violence, or was it the women? I cases (1) and (3), a human being would be able to resolve the ambiguity by employing some common sense reasoning. In case (2), it is a little more difficult, but a judgement could be made (the proposal was rejected yesterday). However, it is possible that only the author of the text really knows what was actually meant. Because a MLIR system typically works with a list of terms, it is unnecessary to be concerned with resolving ambiguities similar to those encountered in examples (2) and (3). MLIR systems need only concern themselves with lexical ambiguities similar to the unclear term 'pen' in example (1).

A machine translation system might approach this problem in a number of different ways. A fully automated system might use a statistical approach, using previous knowledge to guess the most probable solution. Inspired by a similar approach in speech recognition (Jurafsky and Martin 2000), this method is based on Shannon's noisy channel model (Shannon 1948) where a sequence of good text *(I)* goes into the channel, is subject to signal noise, resulting in a sequence of degraded text *(O)* at the end of the communication channel.

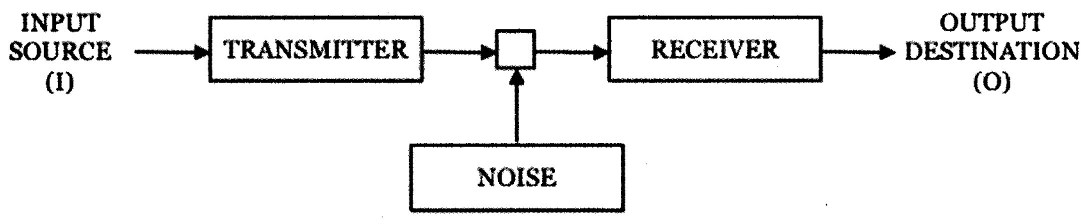

Figure 2.1 – Shannon's noisy channel model

The statistical model for translating texts moves in the opposite direction of the noisy channel model. First, the text considered for translation is assumed to be the degraded output text *(O)* in Shannon's model. The system then tries to "repair" this text, by looking for the most probable input *(I)* (Wilks et al. 1996). To do this, the system maximizes the probability that some text *(I)* is a reasonable translation of the source text *(O)*.

While the statistical model may seem fundamentally flawed at first glance (Noam Chomsky, for example, pointed out that the sentence "'I saw a triangular whale' is highly improbable, but a well formed sentence" (Wilks et al. 1996)), this approach has seen marked progress in recent years. The first successful system to use a statistical approach appeared in the 1970s: the Météo system was developed at the University of Montreal and was designed to translate weather reports (Hutchins 2001) between English and French.

In order to develop a reliable statistical model, this approach relies on aligned parallel corpora of text. Each corpus must represent a direct translation of the other. A good example of such aligned corpora is the Canadian parliamentary debates, which are published in both English and French. However, this is often cited as a fundamental weakness in the statistical approach, since a system requires large parallel corpora in order for it to translate with any degree of accuracy. This is problematic where such large parallel corpora are rare, for example, if either or both the source and target languages are minority languages.

In a MLIR retrieval system, the statistical model can be used to help resolve ambiguities of search terms, however, it may not be considered as reliable as when used with larger texts. The reason for this is that the statistical model requires some knowledge of context in order to be successful. Suppose an individual initiates a query in French using the search term "basse". In French, the word "basse", has a meaning related to sound or to music, so the searcher would only expect the system to return documents of that type. In English the equivalent term "bass" could mean either a musical instrument or a type of fish.

A statistical approach would only be able to resolve the ambiguity when the search term is coupled with another term (co-occurrence) that would add value and give some sense of meaning within a context. For example, the query "bass and fishing" or "bass and performers" could provide the system with enough information to resolve the ambiguous term. While the statistical approach has seen some success in recent years, the current trend is to couple a statistical approach with other methods (Wilks et al. 1996).

Some machine translation systems by design require more participation from a human being in order to gain knowledge about the source text and resolve ambiguities. These systems typically prompt the user, who may need to have some prior knowledge of the original text, to choose a solution from a list of possible alternatives, as in Whitelock's description of how "the system queries the user to resolve ambiguities or indeterminacies until a representation is reached which is sufficiently detailed and precise to guarantee acceptable translation to a given language" (Whitelock et al. 1986). In many cases this method is an acceptable alternative to a fully automated system where relatively high quality results are achieved without requiring expert knowledge in the target language. It is reasonable to apply this method to a MLIR system where the user initiating the query would be able to assist the system in resolving simple ambiguities similar to those found in example (3) on page 14.

An alternative method for overcoming the challenges of MLIR is to use a controlled vocabulary. While a controlled vocabulary restricts the user to a certain degree by only allowing searches with predefined terms, this method is used frequently in library and information sciences to solve problems related to homographs and synonyms. With this approach, each concept is described using one authorized term. Documents are then indexed using this predefined vocabulary, applying the best terms available to describe the document. This method can also be used in MLIR to solve the problems associated with ambiguous search terms. When this is combined with Grefenstette's (1998) theory that the simplest way to find translations in a MLIR system is to use a bilingual dictionary, it is an effective method for producing translations with relatively low complexity.

Jorna and Davis (2001) note that multilingual thesauri can play a significant role in facilitating cross-cultural communication in an increasingly global information society. When applied to information retrieval, the use of a multilingual thesaurus can be a useful way to leverage the simplicity of using a bilingual dictionary as well as the advantages of using a controlled vocabulary to overcome problems of ambiguity. One example of a multilingual thesaurus is the Government of Canada Core Subject Thesaurus. This thesaurus contains a controlled vocabulary of English and French subject terms used by the Government of Canada to categorize and index all its publications. Below is an example of the English entry for "languages" (*Government of Canada Core Subject Thesaurus* 2004):

Languages / Langue

Subject Category:

LN Language and Linguistics

Used For:

Linguistics

Narrower Term:

Aboriginal languages / Langue autochtone

English language / Anglais

French language / Français

Heritage languages / Langue d'origine

Official languages / Langue officielle

Second languages / Langue seconde

Related Term:

Idioms / Idiome

Language education / Enseignement des langues

Plain language / Langage simple

Terminology / Terminologie

Each entry in the thesaurus is listed under a broad subject category. Each term is linked to other terms through a hierarchical relationship of broader terms and narrower terms. Related terms are also included. The use of a thesaurus has the advantage of providing the searcher with the ability to explore the vocabulary through these established relationships. The user can refine their search using narrower terms provided to them, or browse through related terms to find documents of similar interest. This kind of searching is not possible in a simple bilingual dictionary with simple one-to-one translations of search terms.

While a growing number of information retrieval systems incorporate thesauri, many are strictly text-based (Shiri, Revie, and Chowdhury 2002). A small number of interfaces have made use of graphical displays, mainly for the

MeSH Thesaurus. These include TraverseNet (McMath, Tamaru, and Rada 1989), MeSHBrowse (Korn and Shneiderman 1995), Cat-a-cone (Hearst and Karadi 1997) and Visual MeSH (Lin 1999). An example of a graphical browsing interface incorporating the Government of Canada Core Subject Thesaurus will be presented in chapter 4, demonstrating how an interface for a multilingual information retrieval system might be constructed to facilitate searching documents in different languages using a multilingual thesaurus.

2.2 – Human-Computer Interaction and Usability

Usability is increasingly being recognized as an important consideration in the design and development of computer systems. While the role of the end-user was minimized in early computer systems design, over the past number of years the importance of user-centric design has become more prominent. In doing so, the concept of usability has evolved in a number of different research communities, each having its own focus on the subject (Sjöström and Ågerfalk 2004). Much of the research in the area of human-computer interaction has stemmed from research in perceptual, cognitive and environmental psychology as well as industrial design and ergonomics.

One of the most fundamental principles in human-computer interaction is the idea of affordances as coined by James Gibson in his book *The ecological approach to visual perception* (Gibson 1979). An affordance provides an individual with the potential to perform an action within an environment and is directly dependent on the capabilities of the actor. For example, a set of stairs affords climbing to an individual capable of doing so, but it does not allow an infant or any individual incapable of climbing stairs to perform that action. Gibson classifies some affordances as follows: a medium (air), substances (water, solids), surfaces and their layouts (ground, steps, bridges), objects (attached or detached), other animate objects, places and hiding places.

In the late eighties, while a professor of cognitive science at the University of California, San Diego, Donald Norman wrote and published the well-known work *The psychology of everyday things* (Norman 1988). Later republished as *The design of everyday things*, Norman's work has influenced designers in many fields, including industrial designers and designers of computer interfaces (Shalizi 1995). He emphasizes the importance of having a good conceptual model, which allows users to predict the effects of their actions. Because people are spatial and are good at remembering where things are in their environment, Norman insists that designers should endeavour to develop a coherent and concrete conceptual model of their systems. This conceptual model should coincide with the user's mental model of that system. Norman notes that this communication with the user takes place through what he calls the "system image" (See figure 2.2 on page 22). The design model is the conceptual model the designer has of the structure of the system and how it functions. The system image is what is visible to the user and includes the actual interface as well as the documentation, instructions and labels. The user's model is the mental model developed by the user through interaction with the system image.

Throughout his book, Norman outlines his principles of design, which include *visibility*, *affordances*, *mapping*, *constraints* and *feedback*. In a good

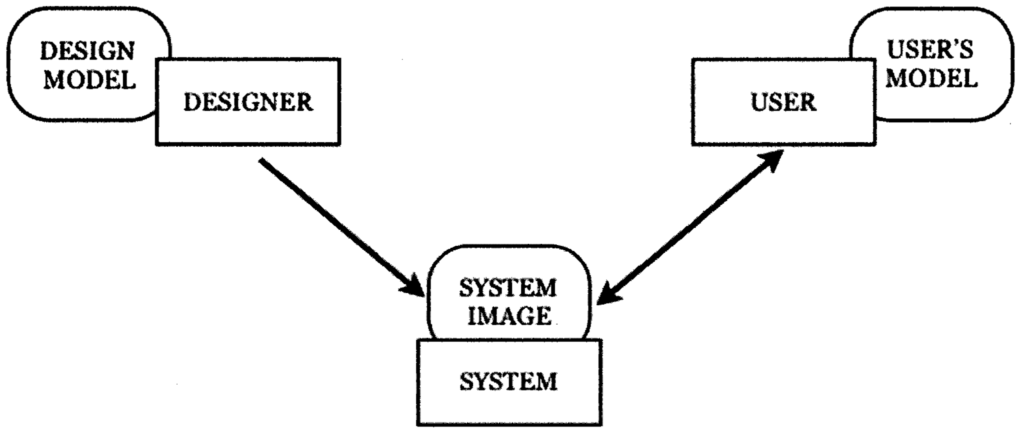

Figure 2.2 – Norman's conceptual model

design, the correct parts must be visible, and they must convey the correct message or allow the correct affordances. When the use of sound is necessary, a system should ensure that those sounds are both clear and audible. In addition, the principle of visibility should not be set aside in favour of aesthetics. Norman points to the example of cabinet door handles that are made invisible or deliberately left out in an attempt to create a more attractive exterior.

While Gibson's definition of affordances includes those actionable properties that are both perceived and not perceived by the user, Norman maintains that designers should care only about those affordances that are visible to the user. He also argues that affordances are not only dependent on an individual's physical capabilities, but also their goals, plans, values, beliefs and past experience. As such, affordances are cultural and must be learned (Gibson and Walker 1984). For example, a kitchen chair can afford a number of actions, including throwing, standing and sitting. While a kitchen chair may be used temporarily as a makeshift stepladder, it is our cultural experiences that dictate to us that the intended purpose of a chair is for sitting.

Controls and displays must make use of natural mappings where the relationship between an action's possibilities and its results are clear. Good mappings often rely on the placement of controls in a way that makes sense. For example, controls on a stove often map to the layout of the burners so the user can easily tell which knob controls which burner (the top-left knob maps to, or controls the top-left burner). Sometimes mappings must take advantage of cultural norms, such as a water tap with the knobs for hot water on the left and cold water on the right.

Constraints limit the way in which an object or device can be used, and they can be physical, semantic, cultural or logical. For example, a large peg cannot fit into a small hole, or text should appear right side up. If a designer is effective in making use of constraints, the user should be able to discover what parts of a device can be operated and which operations can be performed. Problems occur when more than one possible action exists. A constraint reduces the number of possible actions available to the user. If only one action is possible, then there should be no difficulty in performing the action correctly.

Finally, feedback is essential for the user to understand what actions they have performed, enhancing understanding of how the system operates. A designer can offer visual, auditory, or tactile feedback to communicate to the user what actions they have performed and what the system is doing as it processes a request from the user.

In his essay "Design as communication" (2004), Norman develops the notion of design as a conversation between the designer and the user. While not as widely known as some of his more significant works, this short essay is particularly relevant when considering how Habermas's theory might be applied to interface design and human-computer interaction. Norman relates a story of being in a shower in a hotel room and realizing that the designer of the shower was, in essence, communicating to him indirectly through the placement of items like the shower-head and soap dish. The placement of these items was actually telling him where to stand, where to place his soap, and so on. Through the course of his shower, Norman occasionally found himself arguing with the designer.

> And as I prepared to take my shower, I looked back at the soap dish, which was still imploring "put the soap here," and firmly announced "no, I like my soap at the back end of the tub," and I put my newly unwrapped bar there, on the ledge so conveniently provided.

Through this experience, he realized that an affordance may not always be immediately visible to either the designer or the user. For example, the designer may not have realized that the ledge at the back of the tub would be used by someone to hold a bar of soap. According to Norman, since people function through narrative more than through logic, design must be approached as though the designer were telling a story. Furthermore, Norman asserts that a designer communicates a story to the user through the system image, and thus conceptual models are essentially stories.

The metaphor is a long established tool for user interfaces. In *Working with interface metaphors,* Tom Erickson (Erickson 1990) provides some insight into how metaphors can be used to take advantage of a user's existing knowledge to aid in conceptualizing the structure and operation of a system. It should be noted that the concept of using metaphors in computer interfaces predates Erickson's work. The most widely recognized example is the "desktop" metaphor as pioneered by Xerox in the Xerox Star and later popularized by Apple in its Macintosh line of personal computers. This metaphor has been adopted by many other operating systems since Apple first introduced the Macintosh in 1984 and is used in most computer systems that incorporate a graphical user interface, including AmigaOS, Microsoft Windows and a variety of Unix and Linux distributions.

Erickson, however, does provide some insight into how to develop an effective metaphor in the user interface. He emphasizes that poor choice of a metaphor can make it difficult for users to understand how the system is meant to be used. Erickson outlines three steps to developing a metaphor: first, one must understand how the system really works; second, one must identify the parts of the system that might give the users the most difficulty; and third, once an appropriate model has been identified, metaphors that support that model must be generated. There are a number of questions that must be asked when evaluating a metaphor, such as how much structure does it provide? How much is relevant to the problem? Is the metaphor easy to represent? Is it suitable to the audience (i.e. will they understand the metaphor)? Is the metaphor extensible? Once a metaphor has been decided upon, the designers should be consistent with the metaphor throughout the system.

Stubblefield (1998) looks at design metaphors and their influence on the functionality of a system, the methodology of a project and the interactions between members of the development team. He found that design metaphors can be both a benefit and an obstacle to an interface. Upon implementing what was determined to be a strong metaphor, it became difficult to respond to a greater understanding of users' needs and abilities. He also found that a metaphor can break down if the underlying computational complexity of a system is severely different from that of the source.

As the study of human-computer interaction has evolved, many researchers have seen the need for developing systems for appraising the usability of a system. In his book *Usability engineering* (1993) Nielsen provides concrete methods for systematically developing and evaluating effective user interfaces. Nielsen builds on the work accomplished by his predecessors, but his most valuable contribution is his list of ten usability heuristics for evaluating computer interfaces and ensuring their high quality and superior usability. Nielsen's heuristics are as follows: (1) *Use simple and natural dialogue.* Interfaces should be simplified ("less is more") and should tell only what is necessary and ask only questions that users can easily answer. (2) *Speak the users' language.* Designers should make mappings and metaphors culturally appropriate and they should avoid using jargon. (3) *Minimize user memory load.* Computer systems should provide needed information in a timely manner when it is needed. (4) *Consistency.* Terminology and required actions should be consistent throughout the system and its interface. (5) *Feedback.* Keep the user informed about what the user is doing. (6) *Clearly marked exits.* Users should be able to easily escape from unintended situations. (7) *Shortcuts*. Frequent actions should be quick and easy to complete. (8) *Good error messages*. Error messages should be clear, descriptive and given in plain language. Beeps and codes should be avoided. (9) *Prevent errors*. Wherever possible, prevent errors from occurring by keeping choice and actions simple and easy. (10) *Help and documentation*. Provide clear, concise, and instructive online help, instructions and documentation. Using these heuristics, a designer can perform a systematic evaluation to find usability problems in the design of a system's interface.

In *Designing the user interface: strategies for effective human-computer interaction* (1997), Shneiderman outlines a number of principles, guidelines and theoretical foundations to encourage high-quality user interfaces. He introduces the concept of *universal usability* where designers develop interfaces that are universally accessible, taking into account variations of users' physical abilities, cognitive and perceptual abilities, personalities, cultures and physical environments of workplaces. Guidelines with respect to navigation, organization of the display, getting the user's attention and facilitating data entry are presented. Shneiderman insists that information should be consistent, require minimal memory load on the user and should allow for efficient assimilation by the user. The display should be flexible and should allow the user to control the display. The use of colour, intensity, fonts, size of elements, blinking and audio should be subtle except for in times of emergencies, and should be appropriate for the task at hand. Shneiderman encourages designers to know the skill levels of their users and design interfaces with their abilities in mind. Eight "golden rules of interface design" are discussed including: *strive for consistency, cater to* *universal usability, offer informative feedback, design dialogues to yield closure* (actions should be organized into sequences with a beginning, middle and end), *prevent errors, permit easy reversal of actions, support internal locus of control* (users should feel like they are in control of the system) and *reduce short-term memory load.*

In *Designing visual interfaces* (1995), Mullet and Sano describe a number of techniques for improving the visual quality of user interfaces. Six areas where design principles and techniques are presented: (1) elegance and simplicity, (2) scale, contrast and proportion, (3) organization and visual structure, (4) module and program, (5) image and representation and (6) style. Mullet and Sano bring a number of lessons learned from print media into the area of graphical user interface design.

According to Mullet and Sano, elegance and simplicity involve the careful selection of the elements to be included and emphasized. Interface elements must "be unified, refined and enhance the fitness of the solution" and clutter and visual noise removed. This can be achieved by removing redundant features, creating geometric continuity and combining redundant elements for maximum leverage.

Scale means that an element should "fit its space and surroundings". Contrast produces visual differences in "shape, size, colour, texture, position, orientation and movement" of elements in the interface. Proportion deals with ratios, balance and harmony of interface elements and their positioning. Scale, contrast and proportion can be achieved by layering content in such a way so that the user can ignore nonessential elements, sharpening visual distinctions, and establishing a balance between scale and visual weight of visual elements.

The organization and visual structure of an interface can prevent the user "from imposing their own structure, leading to a breakdown in communication". Groupings should be based on Gestalt principles, similar items should be grouped together, and perceptual prominence of interface elements should agree with the intended reading sequence. Organization and structure can be achieved by using symmetry to ensure balance, aligning elements to indicate visual relationships, and through the use of white space to draw attention to important areas of the display.

Module and program refers to the repeated sizes, proportions, forms and ideas within a program that bring predictability and structure to an interface. Systems should also be flexible to account for extremes cases. Common elements should be repeated throughout a program to reinforce themes and modular units should be established through the use of a grid system.

Image and representation are essential for identification, expression and communication. When developing imagery, essential characteristics of a concept or function should be abstracted and generalized, making it easier to be identified. Icons should present either verbal or pictoral representations. If an image needs to have text attached to it to clarify its meaning, the image is obviously poorly designed. If a concept is abstract, words are usually more effective than images. Also, a set of images should "share a coherent language of form that makes their interrelatedness immediately apparent".

Style establishes a distinct, discernible language, providing emotion, connection and context to a design. A design should be distinct, appropriate, and apply to a wide range of artifacts. Designers should familiarize themselves with the published style guides for the environment they are developing for. When developing across platforms, a balance is needed between constancy across environments and consistency with the platform's style guide.

A recent trend in both human-computer interaction and industrial design is to move beyond simple improvements in the usability of devices, and to enhance the emotional appeal of products. Patrick Jordan's *Designing pleasurable products: An introduction to the new human factors* (2002), and Norman's *Emotional design: Why we love (or hate) everyday things* (2004) look both at and beyond usability and explore how users can not only use products effectively and efficiently, but also enjoy using those same products.

In *Understanding computers and cognition* (1986), Winograd and Flores develop the idea that both human thought and language must be taken into consideration when designing information systems. They emphasize that design is the "interaction between understanding and creation" and that "understanding must incorporate a holistic view of the network of technologies and activities". Like human beings, technology operates in the domain of language, inasmuch as it involves the handling of symbolic and linguistic elements.

According to Winograd and Flores, much of what pervades the field of computer science is based on a "rationalistic" tradition. Rationalists believe that language is simply a system of symbols formulated into patterns to represent the world and that cognition is simply the manipulation of these representation structures in the mind. Winograd and Flores challenge the rationalistic tradition, arguing that "we need to replace the rationalistic orientation if we want to understand human thought, language and action, or to design effective computer tools." Their views stem from many ideas found among phenomenologists such as Merleau-Ponty, who has had a significant role in the "anti-cognitivist" movement of cognitive science, and Heidegger who argues that language is much more than syntactically discrete representation of meaning. One of the main insights of phenomenology is that the act of interpretation pervades every aspect of everyday life. Winograd and Flores generalize Heidegger's philosophy into four main points:

> (1) Our implicit beliefs and assumptions cannot all be made explicit.
>
> (2) Practical understanding is more fundamental than detached theoretical understanding.
>
> (3) We do not relate to things primarily through having representations of them.
>
> (4) Meaning is fundamentally social and cannot be reduced to the meaning-giving activity of individual subjects.

While the rationalist believes that knowledge is just a storehouse of representations, Heidegger believes that knowledge lies in the *being* that situates us in the world. This leads into a concept which Heidegger terms *throwness,* or the condition of being in the world. According to Heidegger, it is unusual in ordinary everyday life to make observations, form hypotheses and mindfully choose a prudent course of action. As a result, the designer must accommodate users who will only plan once something breaks down.

Winograd and Flores also explore the concept of perception and how it is understood. They draw on the works of Maturana and his study of the organization of a frog's retina. His findings suggest that we must study perception from within the organism and not from the outside, since perception is dependent not on the perturbant (the object the organism sees), but on the structure of the organism's neurological system itself. Winograd and Flores conclude that the most successful designs are those that are aligned with the fundamental structure of the domain in question, rather than an attempted model of the domain within which they operate. Looking back at Norman's conceptual model (see figure 2.2 on page 22), Winograd and Flores suggest that the system and its corresponding image should correlate as closely as possible. Such a correlation reduces the gap between the designer's conceptual model and the user's conceptual model, increasing successful communication and mutual understanding between the user and the designer, and ultimately resulting in a more confident and productive user.

2.3 – Rhetoric and the Theory of Speech Acts

The study of the art of persuasion has a long and elaborate history. The Sophists of Ancient Greece studied, taught and practised rhetorical techniques to achieve their own purposes. Rhetoric is usually defined as the art or method of persuasion through the use of spoken or written language (Newall 2005). No longer confined to application within political and legal discourse, the study of rhetoric is now being applied to such diverse disciplines as philosophy, literary theory, history, journalism and advertising. The study of rhetoric has influenced a number of theoretical frameworks and analytical devices, including speech act theory, pragmatics and discourse analysis.

The study of rhetoric has produced a taxonomy of principles (canons), modes and devices. The five canons of Classical Rhetoric include, *inventio* (invention, or coming up with ideas), *dispositio* (arrangement, or the ordering of an argument), *elocutio* (style, or how something is said), *memoria* (memory, or how the speaker recalls information through the use of tools such as mnemonic devices), and *pronuntatio* (delivery, or the way in which a discourse is performed) (Cline 2007). In Aristotle's philosophies on rhetoric, he outlines three modes of rhetoric: *pathos* (appealing to the emotions of the audience), *logos* (the use of logic and reasoning), and *ethos* (the credibility or persuasive appeal of the speaker as determined by the audience) (*Silva Rhetoricae*). An author or a speaker may employ any number of rhetorical devices to evoke an emotional response in his or her readers or listeners. Among a sizeable list of devices, these may include the use of expletives, similes, analogies, metaphors, parallelism or hyperbole (Harris 2005). Newall (2005) argues that there is no such thing as too much rhetoric. A speaker can be criticized for employing too much pathos at the expense of logos, or vice versa, but the effectiveness of a discourse is dependent on a number of things such as location, audience and style.

Rhetorical philosophy has influenced a wide range of topics in critical theory, including pragmatics. Many formal approaches to the study of language take as their model the language of mathematics. Such a model presupposes the idea that the truth of a statement can be determined outside the context of a given situation. In real language, however, what is unspoken is as much a part of the meaning as what is actually vocalized (Winograd and Flores 1986). The theory of speech acts deals with the various intricacies of communicative action. Winograd and Flores define the theory of speech acts as the "analysis of language as meaningful acts by speakers in situations of a shared activity" (Winograd and Flores 1986). The social aspect of language is of critical importance and must be considered when analysing a communicative act. Language is indeed a form of social action and must be treated as such, rather than just a mere representation. In order to understand meaning, we must examine this social aspect rather than just the mental element.

The theory of speech acts draws a great deal from Austin's *How to do things with words* (1962). In his book, he introduced his theories about performatives, where an utterance is not considered strictly true or false, but is considered to be felicitous, or appropriate within the context of that utterance (Winograd and Flores 1986). According to Austin, three acts are performed simultaneously whenever an utterance is made: *locutionary*, *illocutionary* and *perlocutionary* acts.

A locutionary act is simply to articulate a phrase with a particular sense and reference, referring to the surface meaning of an utterance. Whenever a locutionary act is performed, an illocutionary act may also be performed. An illocutionary act is any speech act where the remark is directly associated in some way with the actual act itself. These utterances could be acts of warning, stating, questioning, commanding, promising and so on. For example, "He said to me, 'Shoot her!'" is an example of a locutionary act. The utterance "shoot", means the act of firing a pistol, where "her" refers to some female character. In this example, the illocutionary act is the act of urging, advising, or ordering "me" to shoot "her" (Austin 1962).

A perlocutionary act is a speech act that is external to the utterance, yet has the ability to induce some consequential effects upon the hearer, such as persuading, convincing, scaring, enlightening, inspiring and so on (Ljungbert and Holm 1996). It can be thought of as the external effect of the locutionary act. In the previous example, the perlocutionary act, or the consequence of the utterance "He said to me, 'Shoot her!", is that "he" got "me" to, or made me shoot "her".

Austin's student, Searle, later extrapolated on these ideas to propose a "taxonomy of illocutionary acts," which include *assertives*, *directives*, *commissives*, *expressives* and *declarations*. Assertives commit the speaker to the truth of a proposition. For example, with the phrase "Sam smokes habitually", the speaker is asserting Sam's smoking habits and is committing to the truthfulness of Sam being a habitual smoker. Directives get the hearer to do something, like the command "I order you to leave". Commissives commit the speaker to some future action, such as "I promise to pay you the money". Expressives express a psychological state, for instance "I thank you for giving me the money". Declarations occur when the propositional content of the utterance becomes reality, such as "I now pronounce you man and wife" (Winograd and Flores 1986, Searle 1969, Searle 1979).

2.4 – Pragmatics

The theory of speech acts has influenced a number of disciplines, including the area of pragmatics. Jürgen Habermas, a German philosopher in the tradition of critical theory and pragmatics, has become influential in furthering speech act theory. Habermas believes that "everyday linguistic interaction is more pragmatic than just semantics" (Benoît 2001). His theories focus on the social aspect of communication and provide a framework for examining questions of relationships between interlocutors.

Habermas presents a set of "validity claims" that raise the expectations of the responsibility of the speaker for an utterance: *comprehensibility*, *truth*, *truthfulness* and *normative right*. An utterance is comprehensible if the hearer can understand the utterance. The question of truth refers to the factual accuracy of an utterance. Truthfulness is connected to the speaker's sincerity in what is presented to be true. Finally, the question of normative rightness pertains to the speaker's right to make claims based on his or her relationship to the hearer (Benoît 2001).

Habermas's theory places the relationship between individuals at the centre of the act of communicating. Successful communication results in agreement between parties. Participants are not oriented to their own individual successes, so consensus is reached without any form of coercion. A speech act places both obligations and responsibilities between the speaker and the hearer. Each participant holds the other to promises implied in a communicative act and accepts responsibility for any promises implied. Habermas proposes that the quality of the communication can only be guaranteed if there is "communicative symmetry" between all parties involved (Wijnia 2004). Wijnia (2004) summarizes Habermas's requirements for the 'ideal speech situation' as follows:

> All parties involved have equal opportunity to start a discussion, and to bring forward arguments and criticize those of others; there can be no power differences between parties involved, as that might prevent relevant arguments being put forward; all participants should act truthfully towards each other, to ensure that manipulation does not take place (Wijnia, 2005).

All parties not only have equal opportunity to participate in the discussion, but they also have equal responsibility to act truthfully towards each other (Benoît 2001).

When individuals convene for information seeking, they share the common goal of reaching agreement with regard to a validity claim. If these relationships of power are unbalanced in any way, then the ideal speech situation is frustrated. In such a situation, pure communicative symmetry cannot occur and the participants switch to what Habermas calls "strategic action," where one or more parties engages in manipulation of the other party (Benoît 2001). Gerald Benoît points out that "the employment of language, in light of some context, influences the hearer's interpretation of speech. Consequently, the hearer's understanding of truth can be manipulated" (2001). The social relationship between the speaker and the hearer includes "cooperative relationships of commitment and responsibility" (Benoît 2001).

2.5 – Speech Act Theory in Information Systems Technology

While the theories of speech acts have contributed much to fields such as social action theory, pragmatics and discourse analysis, there has been but a small degree of application to the areas of information technology and human-computer interaction. While a majority of the research in human-computer interaction has focused on human perception and cognition, communication theory has also played a role in the discussion. The speech act theory developed by J. L. Austin and J. R. Searle, and later enhanced by Jürgen Habermas, has caught the attention of human-computer interaction theorists including Terry Winograd and Fernando Flores (1986), Pär Ågerfalk and Göran Goldkuhl (1999), and Jonas Sjöström (2002). Their position is that computers are not only designed in language, but are themselves equipment for language, and as a result should create new possibilities for the speaking and listening that we do (Winograd and Flores 1986).

The application of speech act theory to the area of human-computer interaction stems from the understanding that all information seeking interactions by individuals, whether they be human-human interactions or human-computer interactions, can be subject to communication analysis (Benoît 2001). In this context, human-human interactions consist of two or more actors communicating with each other using a computer system as a medium for the exchange of information. Human-computer interactions consist of single individuals interacting solely with a computer system where no other human actors are involved in the exchange of information.

While a great deal of research brings together speech act theory and HCI, a focus on human-human communications has occupied a majority of the work. The first system that specifically applied language-action theories to its design was the Coordinator, developed by Winograd and Flores (1988). They argued that a theoretical basis for the design based on a language-action perspective would help actors in an office setting to become aware of their communicative structures, and as a result create more efficient work structures (Schoop 2001). Speech-Act based office Modeling aPprOach (SAMPO) (Lyytinen, Lehtinen, and Auramäki 1987) attempts to develop a methodology for modeling informations systems in the office as organizational discourses. Configurable Structure Message Oriented System (Cosmos) was a project that tried to develop a specification for a computer-mediated communication system (Bowers and Churcher 1989). In the early 1990s, two systems aimed at business process redesign were developed: Action Workflow (Medina-Mora et al. 1992), which expanded on the Coordinator, and Dynamic Essential Modelling of Organizations (DEMO) (Dietz 1994). MILANO (Agostini, Michelis, and Grasso 1997), a computer supported cooperative work system was developed for managing conversations and actions in teams. Schoop (1998) developed a description language for cooperative documentation systems used in hospitals and other healthcare settings. Jones and Basden (2002) explore how Habermas's action types might influence the design of knowledge-based systems. Rather than look at the higher-level issues of interface design, Jones and Basden look at the lower-level issues of architecture, technical design, and engineering. The common principle shared between these various projects is that the design of a system must inevitably enhance mutual understanding and promote relationships between actors (Te'eni 2006).

Probably the most evolved theoretical framework that brings together speech act theory and information technology is Information Systems Actability Theory (ISAT) (Ågerfalk, Goldkuhl, and Cronholm 1999), (Sjöström and

Goldkuhl 2002). Sjöström and Ågerfalk (2004) claim that "research within the language/action perspective has suggested that theories of social action, semiotics and pragmatics may be useful in order to understand the social and organizational aspects of IT systems." The basis of ISAT is the idea that if communication with information systems is like other forms of communication, then it incorporates a network of social relationships between the speaker and the hearer.

Like speech act theory, the main consideration of ISAT is actions within a social context. While using a system, the users must be able to, through the interface, understand what actions are possible and what they must do in order to perform a successful action. That is to say that the users must understand their obligations. The ISAT framework has defined three types of communication: *user-system*, *user-user* and *designer-user*. User-system communication deals with communicative actions that help the user navigate through the interface. User-user communication involves indirect, synchronous or asynchronous communication between users using a computer system as a medium. This form of communication can be defined by information presented in the user interface that is the result of "what others say" and "what I say" (Sjöström and Ågerfalk 2004). Designer-user communication leads the user to make action decisions based on what information is presented to the user by the system. While the differences between user-system and user-designer may seem somewhat vague, Sjöström & Ågerfalk make the distinction the following way: user-designer communication answers the question of what the user can do (i.e. action possibilities), and user-system communication answers the question of where can the user go (i.e. navigation).

While the theoretical framework for ISAT allows for these different types of communication, its pioneers, Sjöström, Ågerfalk and Goldkuhl, concentrate largely on user-user communication in a majority of their research. A great deal of their work to this date focuses on how to design effective computer interfaces to facilitate communication between users. This thesis, however, will focus primarily on user-designer communication. While it is not possible for human-computer conversations to mimic all aspects of human-human conversations, Habermas's universal pragmatics can play a role in better understanding how to make human-computer interaction more effective. The purpose of applying pragmatics to the area of human-computer interaction is to develop interactive systems keeping in mind Habermas's ideas on strategic action as well as the validity claims of comprehensibility, truth, truthfulness and normative right. The next chapter will further explore how these principles can be applied to the realm of human-computer interaction to create more usable interfaces.

Chapter 3 : Human-Computer Pragmatics

The title of this chapter, "Human-Computer Pragmatics", was chosen by combining two distinct terms from two very different fields of study. First, the term "human-computer interaction" is the study of how people interact with computers. Secondly, the term "universal pragmatics", as coined by Jürgen Habermas in his essay "What is universal pragmatics?" (1998), is the study of the conditions necessary for reaching an understanding between two parties through communication. My proposal for a theory of human-computer pragmatics, therefore, combines theories from both of these disciplines to further the field of human-computer interaction. While one could also think of human-computer pragmatics as an approach to human-computer interaction that is practical, sensible, or realistic, the meaning is intended to be a reference to the philosophical study of pragmatic communicative action and its application to the study of human-computer interaction.

Over the past several decades there has been a great deal of improvement in human-computer interaction. The work of individuals such as Nielsen and Norman has brought a new focus on human factors into the design of computer interfaces, where the needs of the user are placed at the forefront. The purpose of this paper is not to argue for or against any one theoretical framework for developing and evaluating user interfaces, but rather to show how Habermas's theory can provide a new perspective that augments existing ideas and practices. In fact, we will see that there is a great deal of overlap between this idea of human-computer pragmatics and other, more traditional approaches. By approaching the issue of usability from an alternative perspective, the aim is to provide additional insight to the challenge of human-computer interaction.

The motivation for applying pragmatics to the field of human-computer interaction comes from the notion that as people use a computer they not only interact with a device, but they also enter into a process of communicative action with that device. Moreover, the user is indirectly entering into a communicative act with the designer of that system. While the singular term 'designer' is used here and throughout this work, the term 'designer' may in fact denote a large team of individuals, which may include engineers, programmers, graphic artists, software architects and quality assurance specialists, as well as other decision makers such as upper management in an organization. When an application interface is developed, the designer intrinsically establishes a setting for a conversation. Just like Norman's experience in the shower of his hotel room (see section 2.2), the designer is indirectly telling the user what actions can and cannot be performed, what functions are and are not available, where the user can and cannot go, and so on. The user communicates to the system (developed by the designer), what information is needed, where the user wants to go, what tools are needed and how to manipulate objects in the system.

It is important to understand that applying universal pragmatics to the design of information systems and their interfaces does not suppose actual synchronous communication with the designer. These "speech acts" initiated by the designer are not transmitted vocally and do not occur synchronously in real-time. The designer communicates these messages to the user through symbols on a screen, such as text, icons, graphics, layout and other conventions. These symbols can be manipulated by the user to produce more symbols that the user may then continue to manipulate over and over again. This perpetual activity of user action and system response results in a dialogue between the user and the system. On a more abstract level, this dialogue is actually occurring between the user and the system designer. How the user manipulates these symbols on the screen provides clues about the quality of communication that is taking place. Manipulation of the interface alone, however, may not provide a complete picture of what is going on; there may be other indicators provided by the user. For example, some physical or auditory clues may provide more information. Is the user tense or relaxed? Perhaps the user is mumbling to himself, talking his way through each action and reaction. Maybe the user is screaming out in frustration, engaging in a futile war of words with the computer screen.

One must recognize that it is indeed the designer that the user is communicating with and not the device (computer) itself. Is it even possible for the user to communicate with an assemblage of silicon and circuitry? Moderate advances in artificial intelligence unfortunately have not yet made such a situation a reality. Admittedly, user-designer communication is non-conventional and occurs in an abstract fashion. The designer does not actually participate in the conversation directly with the user – synchronously or asynchronously. This is because the messages are not actually transmitted to the designer, who never receives them. The designer communicates to the user, through the system image, a vestige of collective thoughts, ideas, values, communicative aptitude and perceived cultural norms. The user is given an impression that communication is taking place as the user experiences responses to individual actions on the system. Recalling Norman's conceptual model (see figure 2.2 on page 22), these responses are provided to the user by the device through the system image.

3.1 – Why Pragmatics?

If this abstract notion of pragmatic dialogue between the user and the designer is recognized as a reality, then the study of how humans communicate with one another is indeed useful in the study of computer interfaces. To do so, one must move beyond the rationalistic view that language is purely representational. In section 2.2, the views of Winograd and Flores on this subject, as well as the views of phenomenologists such as Merleau-Ponty and Heidegger, were presented. The phenomenologists argue that communication must be viewed as something more than just a combination of syntactically organized symbols and signs. Winograd and Flores maintain that the rationalistic tradition must be set aside in order to not only understand human thought and language, but also in order to develop effective computer interfaces. In order to understand how humans truly think, act and communicate, the rationalistic tradition must be set aside. Habermas's theories on the pragmatics of communication fit nicely within this argument.

In Habermas's theory, the intersubjectivity, or in other words, the relationship between individuals, has a central role (Wijnia 2004). Habermas understands that humans do not converse in a purely rational manner. His theories focus not on semantic primitives, but on how individuals in a social sphere orient themselves as they attempt to communicate (Benoît 2001). In fact, according to Habermas, not all speech can be considered true communication. Habermas outlines two different forms of speech actions: communicative action and instrumental action (also referred to as strategic action).

Strategic actions are goal oriented and are destructive to communicative action. This speech action is exercised when one party views the other as an object or resource rather than an equal participant in a social encounter. While communicative action is supported by the validity claims of comprehensibility, truth, truthfulness and normative right, strategic action is evaluated entirely in terms of efficiency and effectiveness. Strategic action employs manipulative tactics in order to exercise control over the opponent and have him fulfill certain requests, resulting in a hegemonic imbalance between the interlocutors.

Communicative action aims for consensus and is the only action where true communication takes place. Habermas claims that illocutionary acts (commanding, promising, questioning, etc.) and communicative action are intrinsically related. With an understanding that actually-occurring speech never results in true communication, Habermas developed this model of idealized speech in order to better understand and critique real-world speech. It is this same model of idealized speech championed by Habermas that will be used in this paper to establish a framework for critiquing computer interfaces. Benoît (2001) explains how universal pragmatics can be applied to human-computer interaction:

> ...both parties (information system and end-user) are viewed as self-advocating agents, capable of presenting utterances, accepting responsibility for the utterance's comprehension and truth, willing to permit challenges to

> the truthfulness (sincerity) of the utterance, and acting in accordance to normative right.

Human-computer pragmatics, like universal pragmatics, considers individuals, the designer and the end-user as being responsible for each of the four validity claims. Both parties must act according to these claims in order for true communication to take place. However, in practice, the burden is placed largely on the designer to move toward true communicative action and away from strategic behaviour. While this may be seen as a violation of the principle of equal responsibility, the reality is that the designer is primarily responsible for creating the setting for the exchange, the medium by which the communication is to occur and how the user may participate in the conversation. Nevertheless, the user is still responsible for exerting some effort in interpreting the designer's messages.

The idea that the designer takes on more responsibility for communicative action is not out of step with current thinking among experts in the field of human-computer interaction. This is upheld by advocates of user-centred design, such as Norman (1988) and Jordan (2002). At the same time, the end-user is not completely absolved of responsibility in the communicative act. The user must still make an effort to understand how the system works, try to develop a mental model of the system, read dialogue boxes, online help, etc. The user must not expect the system to perform beyond its intended capabilities. The designer must still, however, make an effort to reduce this burden on the user to the greatest degree possible.

While the designer may not be able to carry out a direct face-to-face conversation with the user while the user is interacting with the system, the designer can communicate with a set of potential users during the design process. This can come in the form of surveys, interviews or gathering users into focus groups. Once a product has been shipped, the designer is unable to react and adapt to the communicative actions of the user. In order to obtain feedback from the user before the shipping date, the designer must engage the user early in the design process. The use of storyboards, videos and rapid prototyping tools can be used during consultation with potential users (Caroll 2000). As the prototype evolves into a functional system, users can continue to be involved as they provide continual feedback on a working system. In this manner, the shipped product becomes an artifact of a series of communicative events between the user and the designer.

As communicative features of computer interfaces move toward conformity to Habermas's validity claims, power differences between the user and the system diminish. This occurs as designers reduce their own strategic, goal-oriented actions and place the user's success at the forefront. The result is more comfortable, confident users, whose interests and goals are met. Poorly designed interfaces, on the other hand, widen the hegemonic gulf between the user and the system, resulting in frustrated, less confident users. The result is a communication breakdown between the user and the designer where the user then employs his or her own form of strategic action, attempting to find workarounds, or manipulate the interface in a less efficient manner. In this chapter, Habermas's validity claims will be explored in more detail as each of the four claims is used to critique user-designer communication in computer interfaces. In the following chapter this framework of human-computer pragmatics will be used to critique a prototype of a multi-lingual information retrieval system.

3.2 – Strategic Action

When an individual begins interacting with a system, there are many possible goals the end-user may wish to achieve. The user might be seeking information of some kind, or perhaps is requesting that the computer perform a specific action on their behalf. For example, the user may be performing a calculation, editing a photograph or text document, keeping track of an appointment schedule, or transmitting information to a computer in another location. From the user's point of view, success is measured by the outcome of the interaction with the system. For example, the encounter would be considered successful if they were able to find the precise information they were looking for, successfully verify the spelling in a text document, schedule an appointment, or send an e-mail to the correct recipient. At the same time, the designer is likely to have similar goals. Designers typically want users to be successful while using their products in order to retain their user base and provide income or praise for their organization. Goals which are mutually in harmony with one another do not motivate action that is destructive to communication.

A user may have other goals or expectations that if not met may result in the user employing strategic action. The user may want to see results as fast as possible, perhaps faster than the system is physically capable of providing them.

Some users might have very specific goals, while others may be more vague, perhaps unsure of what exactly they are trying to achieve. For example, when searching for information, a user may want to see very few results, but would like those results to be very relevant to what they are looking for. Others may want as many results as possible that they can peruse and explore, with a hope of finding something worthy of further investigation. Some users may expect a very simple, uncomplicated interface, while others may desire more advanced features. Most users wish to achieve their goals efficiently and with as little trouble as possible, and therefore expect an interface that is intuitive and easy to use.

If the user has difficulty reaching his or her goals efficiently, he or she may begin performing strategic action to compensate for the designer's lack of proper communication to the user about how to perform a particular task. The user may also be forced to compensate for unnecessary constraints built into the system by the designer. Consider, for example, a system where the user browses through a set of keywords linked to documents in a collection. If the system sorts the keywords alphabetically, the user may have to guess how the system has chosen to spell words with alternate spellings, or concepts with alternate names. For example, a user may wish to locate the concept "interdisciplinary", which may also be referred to as "multidisciplinary", "transdisciplinary", or "pluridisciplinary".

Strategic action on the part of the user takes the form of manipulating the interface in a way that was not intended, attempting to find a workaround that will result in the user successfully reaching his or her goals. When this occurs, the user may either be unsuccessful in the attempt and therefore abandon the task altogether, or will succeed in finding a solution after performing a series of tasks through trial and error. Through this series of trial and error, it is possible that the user may discover how to perform the task the way the designer intended. Depending on how long the trial and error process took, it can be said that the designer was more or less successful in communicating to the user how to perform the task. It is also possible that the user employed a longer, more complicated way of performing the same task, and had done so in a series of steps that the designer had not intended for the task to be performed. When a user is found manipulating the interface in this way, the designer has been unsuccessful in communicating to the user how the task is to be completed.

Like the end user, the designer also has a number of goals to reach. Some of these goals may be in concert with those of the user, while others may not. The designer is usually truly sincere in wanting to provide the user with the information, tools or functionality they desire. At the same time the designer may have other goals that interfere with those of the user. It should be noted that designers may not always be aware of the consequences of certain decisions made during the design and development of their systems. The designer may begin to employ certain strategies either intentionally or unintentionally in order to reach certain goals. For example, the designer may have to meet a deadline and may need to make certain decisions that are not in the user's best interest. This could be in the form of a reduced feature set, buggy code, or a poorly designed interface with a general lack of user friendliness. Similar problems may occur if the designer is restricted in terms of funds and needs to make sure the project is completed under budget. As a result of these monetary and time constraints, or perhaps even outright negligence, designers may fail to test their products on all platforms and in all environments where their products will be used. The most common example of this would be a web application that has not been thoroughly tested on all of the mainstream web browsers.

A designer may also have more blatantly egotistical goals when creating a user interface. It is not uncommon for designers to place form at a higher level of importance than function. Some designers might want to make the interface look stylish or trendy. They might want to employ a minimalist theme, when in reality, additional buttons and tools would be more practical. When a designer wishes to make an artistic statement for their own personal gain, either to build their own ego or a personal portfolio, the needs of the user can often be ignored. This problem has the potential to arise with both graphic designers and software developers alike. Programmers might be tempted to exercise a "because I can" mentality, overcomplicating an interface and creating unwanted clutter. One example of this behaviour is the inclusion of animated images and scrolling marquees that were irritatingly overused in many early web pages. In modern web design, using advanced browser features is common. If not implemented with older systems in mind, such features have the potential to cause problems for users that do not have web browsers that support such features. When the importance of style is placed ahead of function, the usability of an interface can be greatly affected, reducing productivity and prohibiting users from feeling relaxed and confident. Unfortunately, many designers neglect to take into account how stylistics within their designs might affect the usability of their systems.

It should be noted that not all usability/communication issues are directly related to strategic action. There are often elements that are beyond the control of the designer. While computers have come a long way in terms of their graphical capabilities, designers can be limited in their implementations by existing technology. At the same time, a designer may be reluctant to use certain technologies that are available to them and that would enhance the usability of the system, fearing that a large portion of their user-base will not have the hardware or software requirements to handle more advanced graphics or other sophisticated technologies. The job of the designer is not easy. There are so many different kinds of users, operating systems, web browsers, skill levels and user preferences, and it is often a great challenge to address all of these concerns simultaneously. However, as designers come to understand how their decisions can affect users and their ability to be both happy and productive as they work, they are then more capable of making good design decisions, reducing the hegemonic imbalance that exists between themselves and their users.

3.3 – Comprehensibility

According to Habermas, "the speaker must choose an intelligible expression so that speaker and hearer can comprehend one another" (1998). Comprehensibility addresses the semantic and syntactical concerns of an utterance. As the contributing parties ensure that their statements are intelligible and understandable, the validity claim of comprehensibility is fulfilled. Benoît (2001) suggests that in an information system, messages transmitted to the user must be unambiguous. This can be established through proper use of natural language in labels, dialogue boxes, documentation and help files. Nielsen (1993) counsels designers to use simple and natural dialogue, ensure that exits are clearly marked and that error messages are meaningful and understandable. Many computer users have probably seen a message similar to "System Error 0x50001008 (57493407853). Unspecified error." This would be a good example of an incomprehensible error message. System messages should help the user make informed decisions about to how to proceed. In addition to Nielsen's suggestions, choice and design of semiotic elements as well as proper layout are all factors that contribute to the comprehensibility of a system. Interfaces should also be free of clutter and unnecessary distractions. Graphical elements, colours and tools should not divert the user's attention from the task at hand, and should be considered in terms of how much value they add to the overall functionality of the system.

Software bloat (also known as feature bloat, or bloatware) can be destructive to the comprehensibility of a user interface. Software bloat is a term often used to describe the tendency for systems to contain more features and larger amounts of system resources without consequent benefits to the users. It is often difficult for users to understand how to perform a task if a system contains a massive number of features and commands. It is particularly difficult to understand how to perform a task if the interface is cluttered with an excessive number of buttons, menus and controls. Feature bloat has a number of causes, many of which are not directly related to careless design. Software companies often feel the need to keep adding more features to stay ahead of their competition, or to encourage their customers to upgrade to a newer, more powerful version of their software. Rapid development cycles and market-driven deadlines can put pressure on development teams, preventing even the most meticulous designer from performing effectively. Adding unplanned features to the product late in its development cycle can cause a project to drift from its original design goals, contributing to unwanted clutter and inconsistencies in the design, resulting in a less intelligible system.

3.4 – Truth

Perhaps the importance of truth is most obvious compared to the other three validity claims. If a program does not produce accurate results, it will be of no benefit to the user regardless of its adherence to other fundamental principles of design. First and foremost the system must be bug-free and devoid of systemic errors. Thorough testing through a quality assurance process is essential if a system is going to perform correctly. Online help and other documentation should be up to date and instructions, examples, screen shots and other references to the interface must match the current version of the system. Generally, the system should provide information that is verifiably true.

In terms of information retrieval, the question of truth can be determined by the set of retrieved items and whether or not they are about the topic of interest. In many instances it is likely that the system's understanding of what is true and correct differs from the user's understanding of what is true and correct. One of the greatest challenges in information retrieval is finding the perfect balance between precision and recall. The system must provide the user with the precise results that the user will find relevant without inundating the user with items that are completely irrelevant (low precision, high recall), or conversely, neglecting items that the user may find relevant (high precision, low recall). Improving the relevancy of search systems is nontrivial and often necessitates the development of elaborate computational algorithms.

A comprehensive discussion of the intricacies and complexities of the scientific study of information retrieval systems will not be explored here. However, it should be noted that most algorithmic approaches to information retrieval result in a hegemonic imbalance between the system and the user. The system decides on behalf of the user which items are weighted more heavily on the side of relevance, which items are considered to be less relevant, and which items are considered to be completely irrelevant. This is largely due to the great challenge of defining relevance and implementing it in relevance ranking systems. Such systems imply that the *system knows best*, rather than the *user knows best*. The next chapter will explore some ways to empower the user through a browsing strategy, giving the user the tools necessary to make their own decisions about relevancy. As the user browses, groups and sorts collection items, the system is able to display to the user precisely what they have asked for without having to incorporate a complex algorithmic solution. Such systems instill in the user a greater degree of confidence that the results are accurate. Engendering such confidence in the user helps the system conform to the validity claim discussed in the next section: *truthfulness*.

3.5 – Truthfulness

While the difference between the validity claims of truth and truthfulness may not be immediately apparent, the former is the accuracy of an utterance, whereas truthfulness is the perceived sincerity of an utterance, or, that is to say, is that the speaker's intentions are recognized and appreciated for what they are. The question of truthfulness should be addressed with the understanding that the end user enters into a relationship of trust with the system and its designers. Certainly *truthfulness* is somewhat dependent on the conformity to the validity claim of *truth*, but absolute conformity to *truth* does not guarantee complete conformity to *truthfulness*. In order to conform to the validity claim of *truthfulness*, the designer must be able to convey a sense of sincerity and certitude that the information presented to the user is indeed correct. If the information presented to the user is true, but the user does not feel confident that that information is indeed correct, the designer has failed to establish truthfulness in the design. One way a designer can avoid such a pitfall is to avoid strategies that cause the user to feel as though the system is hiding information from the user. Essentially, designers should find ways to make their systems more trustworthy, instilling within the user a feeling of confidence that the system is performing its functions successfully and correctly and is providing the user with accurate information.

Adherence to Nielsen's feedback heuristic (see section 2.2) is a good place for designers to start as they strive toward truthfulness in their systems. Doing so will indeed foster a greater degree of trust by their users. Many systems provide some kind of feedback, but simple indicators that the system is busy performing some task is not always enough. The user wants to know *what* the system is doing, so more information may be necessary. A simple animated hourglass, spinning dial, or progress bar does not provide the user with sufficient information to understand how the system works and what it is doing. An information retrieval system could, for example, show an animation of the documents being separated out of the main collection, sorted and placed in some logical sequence. In some situations, a simple textual description of the action being performed may be sufficient. For instance, if the system is attempting to connect to a remote server, it is not particularly useful for most users to see an animation of network packets as they travel over a network (although this would be a great way for more advanced users and network administrators to debug connection failures and other network issues).

Norman, in a way, promotes truthfulness as he advocates an effective depiction of a system's architecture through the "system image", allowing the user to develop an appropriate conceptual model of that system. While a good conceptual model helps users to successfully perform tasks, an understanding of how the system works will also help them be more confident in the information provided by the system. The majority of current information retrieval systems provide poor conceptual models for their users. Many do not provide the user with any sort of information about what happens between submitting a query and displaying the results. Users may begin to wonder if the system is returning everything relevant to their query, or if some documents are missed. They may wonder if the results shown really are the most relevant, or wonder if the results are actually listed in the order that is most relevant to them rather than what the system decides is most relevant. If users do not know how search terms are being used to provide results, they may not understand how to refine their queries to provide more relevant results. In extreme cases, users may begin to believe in conspiracy theories, suspecting that certain results are either ranked lower or not shown at all due to certain political or financial motivations or some other self interest on the part of the creators of a search engine.

In addition to providing a better conceptual model of the system image, information retrieval systems can improve truthfulness as they obtain feedback from the user regarding the quality of the search results. One strategy often employed is the inclusion of a "relevance feedback" feature. The idea behind this approach is to take the results returned from the initial query and use information provided by the user about whether or not those results are relevant, and then perform a new query based on this feedback from the user. The information can be gathered by providing the user with an opportunity to look for more documents similar to a given result (i.e. "more like this").

3.6 – Normative Right

The validity claim of normative rightness deals with larger social concerns as humans interact with computer systems. According to Habermas, normative rightness is achieved as interlocutors come to an understanding with one another, primarily as they use words that both individuals can agree upon. This particular claim can be very difficult for designers to deal with, especially those who are developing systems that will be used by a very large user base. It can be a challenge to anticipate every user's technical, linguistic and cultural background, ensuring an inclusive interface that is not offensive to potential users.

Many software companies, especially those that develop products used by individuals around the globe, struggle with the challenge of catering to a large, linguistically and culturally diverse user base. In its recent revision to the company's flagship product, the Windows operating system, Microsoft was confronted with this challenge as they attempted to address one cultural sensitivity in a small application. Since 1992 Microsoft has included a simple game called *Minesweeper* with its operating system. The essence of the game is for the user to successfully navigate a minefield, discovering all the hidden mines without having one explode. Recently, Microsoft considered altering the game substantially to deal with the ongoing criticism that the nature of the software lacked sensitivity toward individuals and nations that have been or continue to be adversely affected by land mines. Microsoft's solution was to reengineer the game so that it could support different "skins", allowing the game to work with either mines or some other arbitrary object such as flowers. The designers were then faced with the challenge of having to change the name of the game, the text in the help file, the use of the word "explode", in addition to a number of other significant alterations. Throughout the entire process, Microsoft had not anticipated how seriously people were taking this issue. Some countries felt that these changes were not sufficient, and that they didn't even want to have the *option* to switch to mines. One of the developers, in expressing his frustration with the complexity of the issue, sarcastically proposed that the name of the game be changed to "Microsoft Windows Vista Logic-Based Hidden Item Seeking Game 2006 with Skins" (Vronay 2006).

Adapting to cultural concerns can be a challenging task for designers. Calendars and counting systems as well as acceptable colour schemes and patterns vary between cultures. Designers must consider their audience as they make decisions with respect to language, metaphors, colour, icons and other graphics.

> In most projects, the complex interplay of user, business, marketing, and engineering requirements needs to be resolved by Web user-interface and information visualization designers. Their development process includes iterative steps of planning, research, analysis, design, evaluation, documentation, and training. As they carry out all of these tasks, however, they would do well to consider their own cultural orientation and to understand the preferred structures and processes of other cultures. This attention would help them to achieve more desirable global solutions or to determine to what extent localized, customized designs might be better than international or universal ones (Marcus and Gould 2000).

Many large-scale software projects include a "localization" process where a slightly altered, custom version of the software is created for use in certain regions. This process typically occurs at the end of the development process and is often limited only to a translation of the software into some target language or languages. The drawback of this approach is that many culturally dependent and culturally sensitive decisions are made long before localization takes place. It is often difficult to address issues of culturally incompatible metaphors at the end of the development cycle. Ideally, the localization team would be involved earlier in the process, throughout design and implementation, not just at the end. As this team becomes more involved in the initial design process, a more diverse set of cultural sensitivities should be considered, not just those of a linguistic nature.

Culturally sensitive localization is not the only concern designers have as they strive toward normative rightness. Designers must not only be culturally aware, but also lexically aware as they think about those who might use their applications. The language employed in software used exclusively by a specific group should match the accepted vernacular of that particular group. Nielsen's second usability heuristic advises designers to "speak the user's language". For example, software that is designed for and will be used exclusively by typographers should employ language that typographers use and expect. The terms "kerning" and "leading" should be used to denote spacing between characters and lines respectively. On the other hand, if the software is to be used by a general-purpose audience, the principle of comprehensibility as discussed in section 3.2 should be observed. For example, a general-purpose word processing application should avoid using such jargon and instead opt for more generic and descriptive terms like "character spacing" and "line spacing".

Recent attention on "Universal Usability", developed by Ben Shneiderman (2000), recognizes the diversity of users and their needs. Stemming from the concept of *Universal Design* (also called *Inclusive Design*), it is based on the understanding that there is no such thing as an "average" user and in many cases a single system cannot always accommodate everyone. Its exemplar, Universal Design, was originally conceptualized by architect and designer Ronald Mace and aims to design products usable by all people (Instone 2004). Adapted specifically to the scope of information systems, Shneiderman's Universal Usability strives to provide computer systems that are usable by all citizens.

When an interface design cannot meet the needs of a large segment of the user population, multiple versions of the system, or adjustment controls should be made available. Novice users can be presented with only a few options initially. Over time, as the user gains confidence, the user can alter the interface to include more advanced features. Shneiderman together with Harry Hochheiser have developed what they call a "Universal Usability Statement Template", which designers can use to state a Web site's content, browser requirements, network requirements and other characteristics that may impact its usability (Hochheiser and Shneiderman 2001).

As with Habermas's universal pragmatics, human-computer pragmatics insists that both parties accept responsibility for conforming to normative rightness. Recently in an article published in *Interactions,* Norman cautions designers not to be overly user-centric in their designs, and instead encourages them to be what he calls "activity-centric" (Norman 2005):

> Builders and designers often have good reasons for the way they constructed the system. If these reasons can be explained, then the task of learning the system is both eased and made plausible. Yes, it takes years to learn to play the violin, but people accept this because the instrument itself communicates rather nicely the relationship between strings and the resulting sounds. Both the activity and the design are understandable, even if the body must be contorted to hold, finger, and bow the instrument.

Here Norman is not renouncing everything he has said in the past regarding user-centred design, however, he acknowledges the reality that the end user must also assume some responsibility for adapting to conventional language and procedures. This includes learning the language used by the system and the set of commands understood by the system. For information retrieval systems, this might be a query language, or a set of tools used to refine search results. For a word processor or a graphics editor, this might include labels, help files, or the set of commands found in the menu system. If a user intends to use an expert system that is designed for a specific group of professionals, the user must be willing to invest the time to become familiar with the trade and the terminology that is used. If the user understands the activity, then the system is understandable (Norman 2005).

True communication is achievable as long as the end user is willing to spend some time learning both the terminology used by the system and the set of commands accepted and understood by the system. The designer, however, is primarily responsible for reducing the amount of time the user needs to spend learning system commands, terminology and how to perform basic tasks.

3.7 – Summary

This chapter has proposed an approach to human-computer interaction based in communicative action theory. Human-computer pragmatics is a product of combining theories from two distinct fields: human-computer interaction, and universal pragmatics. It is the philosophical study of the application of pragmatic communicative action to the area of human-computer interaction. The purpose of this approach to human-computer interaction is to augment existing ideas and practices, using universal pragmatics as a new lens for viewing the field from an alternative perspective.

Human-computer pragmatics views computer interfaces as a platform for conversation. This dialogue takes place between the designer and the end user through the application interface. While the designer is primarily liable for creating this platform for communicative action, both parties undertake certain responsibilities as they meet to engage in dialogue. The designer communicates certain messages to the user including various thoughts, ideas, values and cultural norms through the various semiotic elements on the screen. The user engages in the dialogue by manipulating these graphical elements.

The application of Habermas's universal pragmatics to the area of information technology adopts Habermas's two classifications of speech acts: communicative action and instrumental, or strategic action. Designers can create more effective user interfaces if they are able to recognize their own strategic, goal-oriented actions and place the success of the user before their own egocentric goals. If designers instead resort to strategic action as they develop an application interface, power differences between the user and the designer are established, resulting in interfaces of inferior quality and users who are less confident and ultimately more frustrated.

As designers learn how to effectively communicate with their users, their interfaces inherently become more useable. Effective communication can be established through conformity to Habermas's four validity claims: comprehensibility, truth, truthfulness and normative right. Designers must empower those who use their systems, eliminating power differences between themselves and their users. In the next chapter, this will be explored in more detail through the evaluation of a proposed multilingual information retrieval system, using Habermas's validity claims as a guide.

Chapter 4 : Evaluation of a Visual Interface

Due to the numerous advances in information technology over the past few decades, the amount of information that is readily available has increased astronomically. Individuals, and on a larger scale, entire nations and even economies, are becoming increasingly dependent on information. As our society adapts to these new conditions, it is more important for individuals to be able to access information quickly and easily in order to remain competitive. The aspect of usability is one of many important concerns in the design and development of Information Retrieval (IR) systems. While it is now more convenient to access many different kinds of materials than it has been in the past, architectural improvements of IR systems have not always been attended by advancements in the usability of such systems.

The design of IR user interfaces brings a number of challenges over and above the architectural, organizational and algorithmic factors. There exists a great chasm between the language of computers and the language of human beings, which interferes with faultless communication between the two. Computers consist of silicon and complex electronic circuitry and process information in a precise, binary process, counting on logic and set protocols to perform functions and channel information. Conversely, humans communicate in a more complicated, imprecise manner, relying on semiotic systems, social and cultural norms and visual and auditory cues. Computer interfaces are the key to their successful operation, acting as an interpreter on behalf of both the computer and its sentient user, translating between precise, structured protocols of computer systems, and the imprecise, social and cultural aspects of human communication. As with any user interface, IR user interfaces must be able to bridge this gap between the user and the system.

Accompanying current improvements to information access, recent advances in technology have brought a number of significant challenges. One of these challenges that is of particular note is that of globalization. Increases in both worldwide communications and the publishing of information from around the globe have amplified the demand to overcome language barriers between individuals and communities. Despite an ever-shrinking global community, language barriers in many instances continue to prevent the exchange of a great deal of information, ideas and experiences. Information systems must be able to assist individuals as they attempt to cross these language barriers if we are to be truly successful in bringing communities together.

Research communities in particular would benefit a great deal from being able to access and share information on a global level. Doing so could reduce a considerable amount of redundancy as well as encourage a greater degree of collaboration. In recent years there has been increasing interest in the development of information retrieval systems that cross language barriers. Multilingual information retrieval systems allow users to obtain information recorded in a variety of languages by entering queries in a single system using a single language. Individuals are able to retrieve documents relevant to their queries regardless of the language of those documents.

The purpose of this chapter is to explore how to effectively facilitate searching multilingual information repositories. A prototype of a thesaurus-based browsing interface for a multilingual information retrieval system is presented. Certain aspects of the design of the system structure will be discussed, keeping in mind how such architectural decisions affect the usability of the system. The design of the interface and its individual features will be considered with reference to usability principles introduced in chapters two and three. This discussion will be accompanied by analysis using human-computer pragmatics as a framework for assessing the interface.

4.1 – Thesaurus-Based Multilingual Information Retrieval

Over the past decade or so, information retrieval has become a hot topic. Entire companies have been formed to help Internet users navigate and locate information on the World Wide Web. Desktop search has become extremely popular and software vendors are incorporating sophisticated tools in their operating systems to aid users in finding information on their personal computers quickly and easily. These systems typically use a full text search approach. In this case the computer examines all of the words in each document and automatically generates an index or concordance for all documents. When the user executes a query, the index is consulted, an algorithm performs a calculation based on word frequency and other metrics, and a list of results is returned to the user. This method works well in applications like Web and desktop search where the repository is very large or is in a constant state of flux. In contrast, a repository that is more manageable, usually due to its smaller size, often makes use of a controlled vocabulary. In this case items are tagged and indexed, usually manually, according to a predefined set of keywords that describe the contents of the document. These tags are then consulted by the system when a search is initiated in order to find and return to the user a set of appropriate documents.

Many information retrieval systems are beginning to incorporate thesauri to assist users in choosing suitable search terms and expanding their queries. While thesauri can be incorporated into both full text search systems as well as those that use a controlled vocabulary, this strategy is more suited to the latter approach where a set list of terms is used to tag and index documents. In this case, the list of controlled keywords can be mapped directly to entries in the thesaurus. The system can then leverage the connection between related terms in the thesaurus to aid the user in finding the most appropriate term for their search. As the user navigates through connected terms, the system displays those documents that have been tagged with the given term.

Thesauri can be extremely useful when applied to multilingual information retrieval systems. One of the challenges of multilingual information retrieval systems is how to represent search terms in various languages. Grefenstette (1998) suggests that the use of bilingual dictionaries is the simplest tool for finding translations in a multilingual information retrieval system. While translating a query is a much less difficult task for computers to perform when compared to natural language, there are still problems of ambiguity that must be resolved. Information retrieval systems have been using controlled vocabularies for many years as a means to overcome the problem of ambiguity. In addition to aiding users in finding the right terms for their queries, multilingual thesauri in information retrieval systems combine the advantages of controlled vocabularies with a bilingual dictionary to resolve ambiguities and find accurate translations of search terms.

4.2 – Prototype of a Multilingual Information Retrieval System

While many IR systems have started to incorporate thesauri into their design, many are strictly text-based, are available only in advanced search modes, or are completely disjointed from the query building process (Shiri, Revie, and Chowdhury 2002). The prototype introduced in this chapter draws on a number of strategies involving a browsing interface incorporating matrices. A combination of graphically delineated spaces, while dedicated to specific tasks, works together to form a useful environment to assist the user in fashioning multilingual queries. These spaces include a thesaurus space for browsing and navigating the thesaurus, a query space for formulating search statements and a document space for viewing a list of retrieved documents. The mock-up interface is shown in figure 4.1 on page 73. While development of a functional prototype is currently underway, this project consists strictly of an evaluation of the sketches. The numbers shown in this graphic are not a part of the interface, but are for identification purposes only.

The prototype presented here is the result of a combined effort between four individuals from the University of Alberta: Stan Ruecker, Ali Shiri, Ximena Rossello and the author. The primary focus of this research was to investigate how to successfully integrate the thesaurus into the process of building and executing search queries in a multilingual environment. Access to the thesaurus must be seamless while providing ample control to the user as queries are formulated and submitted. Before developing the prototype interface, an example thesaurus was needed. The Government of Canada Core Subject Thesaurus was selected because of its availability, comprehensive lexicon, and overall quality. The Core Subject Thesaurus is well established and is currently used by a number of government agencies for indexing publications and other materials. In addition, and most importantly, the thesaurus is bilingual (French and English). It was developed according to the *Guidelines for the establishment and development of monolingual thesauri* (ISO-2788 1986) and the *Guidelines for the establishment and development of multilingual thesauri* (ISO-5964 1985).

The Core Subject Thesaurus consists of nineteen broad subject categories, each term being classified according to these categories. Every entry is indicated as either a "preferred term" or a "non-preferred term". Documents are indexed only with preferred terms. Non-preferred terms are linked to an alternate preferred term. This is to aid users in finding the appropriate terms when either searching the repository or indexing documents as they are added to the repository. A preferred term in the thesaurus is linked to other preferred terms through various relationships. Hierarchical relationships are indicated through the listing of "broader terms" and "narrower terms", associative relationships are indicated through "related terms". Translations, or "inter-language equivalence relationships" are also indicated.

Figure 4.1 shows how subject categories can be browsed in multiple languages by selecting one of nineteen categories on the left side of the screen (1). This can be used as a starting point for the user to initiate navigation through the

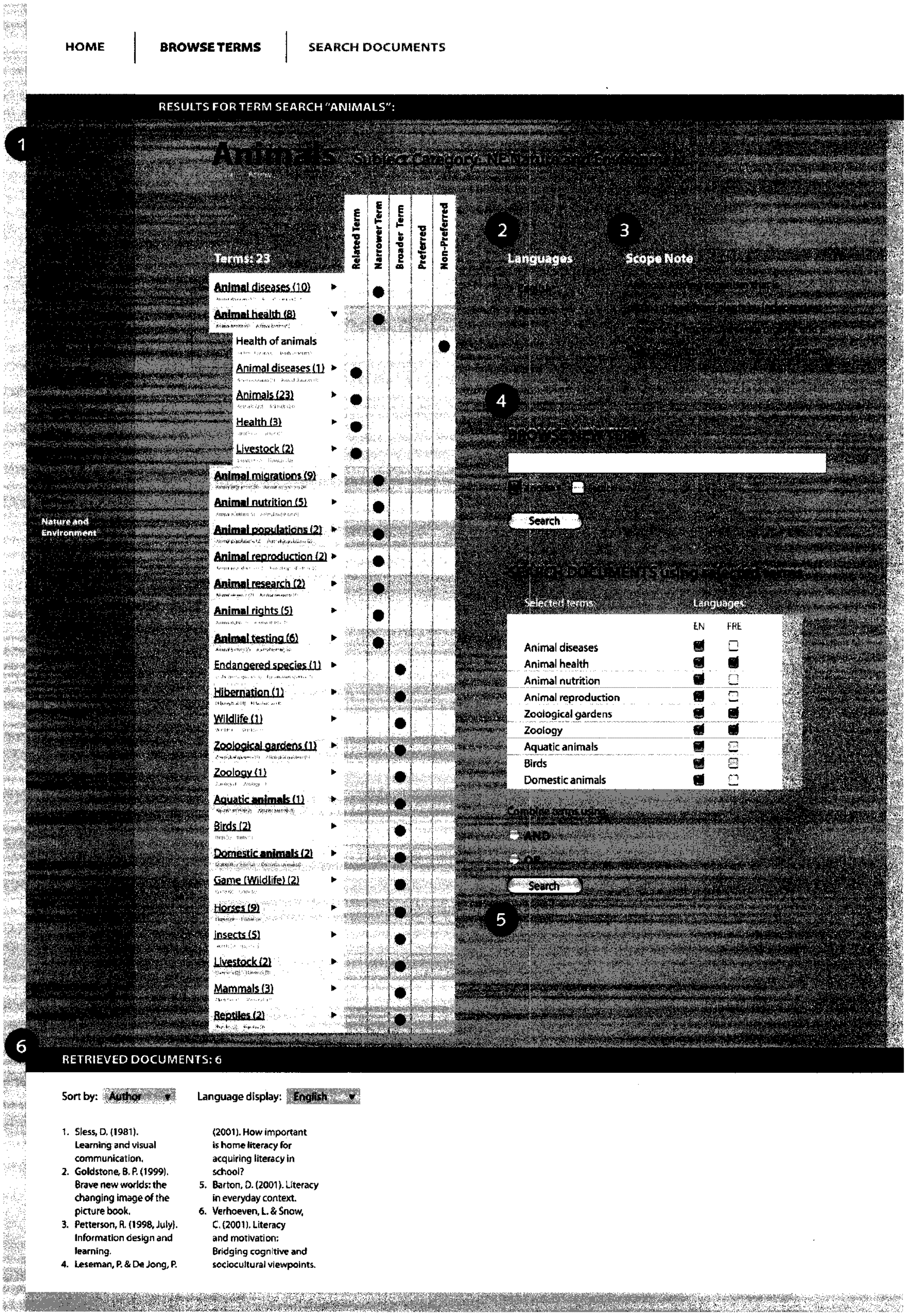

Figure 4.1 – Visual interface for a multilingual thesaurus

thesaurus, or a term can be entered in or searched for using the "Browse New Term" feature (4) located just to the right of the thesaurus table. When a term is found, its entry is displayed in the thesaurus space located in the centre-left portion of the screen. A scope note (3) is offered to provide a detailed textual description of the currently displayed term.

In this case, the primary entry currently displayed is the term "Animals", listed under the "Nature and Environment" subject category. The matrix is used in the thesaurus space to display all existing relationships between the term "Animals" and other terms in the thesaurus. All terms that are linked to the term "Animals" are displayed down the left-hand column of the matrix and the relationship types are listed at the top of the matrix. A relationship is shown by a round dot at the intersection of the row of a given term and the column of a relationship type. This graphic indicates whether the term is a related, narrow, broad, preferred or non-preferred term.

This interface is able to leverage the power of the multilingual thesaurus in a way that makes the information easily accessible to the user as queries are formulated. Each associated term listed under a given entry can be expanded to view that term's own associations. This provides the user with the ability to quickly browse through relationships without having to navigate away from the currently displayed primary term. The user can then choose to expand the search using a broader search term, or narrow the search down using a more specific term.

As the user browses the thesaurus, terms can be selected to be included in a search query. The user can select the terms to be included by clicking the check boxes on the right-hand side. A selector is also provided so a search can be performed using either a combined Boolean *AND* or a combined Boolean *OR*.

Once the user has formulated and executed a query, the system displays the results in the document space (6) located at the bottom of the screen, spanning the thesaurus space on the left and the query space on the right. Additional controls are provided for managing these results, providing a means for sorting the result set. The documents are represented using standard bibliographic information about the author, title and date. These representations also serve as a link to the actual document.

The interface provides a number of ways for working with multiple languages, including altering the language of the text. In the thesaurus space, a microtext below each term displays the term in each available language. This may be difficult to see in this graphic, however, the intention is to dynamically magnify this text when the user places their mouse cursor over the text in order to make this text more legible. This persistent presence of the term's different language alternatives reminds the user that more than one language is available. This also provides a convenient way to switch to a different language. In the panel just to the right of the thesaurus table, another tool is available to the user (2), providing a means for explicitly selecting the desired language. Due to the limited number of languages available in the Government of Canada Core Subject Thesaurus, this prototype offers just two languages: English and French. Nevertheless, the interface incorporates a flexible design, allowing additional languages to be added without crowding the interface or in any way compromising its utility.

4.3 – Evaluation

Over the next five sections the discussion will be based on the principles of human-computer pragmatics as explained in the previous chapter. This will include a discussion of strategic action as well as the four validity claims of comprehensibility, truth, truthfulness and normative right. It should be noted that since the author participated in the development of the prototype, this evaluation is not purely objective. As a result, the author may not be entirely aware of how certain decisions in the design process may affect the user. This may be particularly true in the discussion on strategic action. At the same time, some time has passed since the interface was last visited, so the assessment that follows was executed with a fresh perspective.

Because a completely functional prototype is not currently available, the system being assessed in these next few pages is entirely static. User actions and corresponding system behaviour exist only in conceptual form, and therefore can only be assessed accordingly. Graphical elements, layout, etc. will be considered as they appear in the prototype shown in figure 4.1. User actions and system implementation will be considered as they have been conceived. Because the system has not been implemented, user testing was also not possible. As a result, it is impossible to discern exactly how effectively users would communicate with the system. Therefore, this evaluation will primarily consider the features of the designer's communicative actions, or in effect, the designer's "utterances".

4.4 – Strategic Action

In the previous chapter, the discussion of strategic action led to a number of ideas about how designers might consciously or subconsciously prevent true communicative action through a variety of decisions made throughout the design process. Tight deadlines, lack of sufficient funds and poorly trained personnel all contribute to poorly designed interfaces and buggy code. Excessive, trendy stylistics, an excessive feature set, or even the lack of a sufficient feature set also contribute to a hegemonic imbalance between the user and the designer, resulting in a less functional and useable system.

The design process used in producing this prototype was not immune to the presence of strategic action. The design team consisted of a small group of four individuals including an information scientist (Ali Shiri), a usability expert (Stan Ruecker), a graduate student in art and design (Ximena Rossello) and a graduate student in modern languages who also has a background in software development (the author, Karl Anvik). This group of researchers assembled with a general idea of developing an interface for a thesaurus-based multilingual information retrieval system. The development process that produced this prototype was iterative and somewhat informal. The team members would convene, and discuss various ideas based on previous research in the area as well as each person's individual knowledge and experience. Sketches were made which were used in producing a graphical prototype. During following meetings this prototype would be critiqued by the group, more ideas would be presented, more sketches would be produced and the process would repeat itself, to eventually produce the current design.

While this design process kept a user-centric focus, the user was noticeably absent from the initial process. As is the case with many software projects, the designers often assume that they are an accurate representation of the average user, so many design decisions are made with the thought of how they themselves might find the system useful and useable. If one were to evaluate the hegemonic balance between the user and the designer at this point in the development process, it would appear that the designer has established control over every aspect of the communicative process. However, plans are currently underway to perform a user study of the working prototype after it is implemented. As this process unfolds, a more accurate appraisal of the power differences between the user and the designer will be possible, including their respective strategic actions.

4.5 – Comprehensibility

According to Habermas, comprehensibility is attained when both interlocutors are able to understand what the other is saying. Recall from the previous chapter that if a system conforms to the validity claim of comprehensibility, that system must be intelligible. Language must be unambiguous, and the interface must be free of clutter and consistent throughout the system. Appropriate use of colour, white space (i.e. negative space, or unmarked portions of the screen) and other graphical elements must be employed to make it easy to locate and understand interface elements. In addition to the interface, the help files and other documentation must also be both accessible and comprehensible.

Since the prototype consists of a single screen, consistency can only be evaluated within the single page. In this case, font usage, colour, and other graphical elements appear to be consistent. The language in this interface is both clear and concise. Interface elements are labelled appropriately and descriptively. The use of colour is appropriate, using a combination of shades of green, white and black, contrasting where needed, as in the instance of light text on dark background. Different shades of colour are also used to separate interface elements, such as the category list from the thesaurus grid. The grid also uses shading to show rows and columns and where they intersect.

White space is used to separate sections of the interface with different purposes, such as the browse space (1 in figure 4.1) and the query space (4 and 5 in figure 4.1). White space effectively keeps the interface from appearing cluttered and cramped. The prototype contains an absolute minimal number of features to keep the interface free of clutter. In the query space a design decision was made not to include a Boolean NOT operator in order to simplify both the implementation and the interface. It was decided that the NOT operator would confuse the user more than it would be helpful. It was determined that the Boolean AND operator would be sufficient for filtering and narrowing down the result set.

The prototype, however, is not without its shortcomings. First of all, there is no visual indication that help files and other documentation have been considered. Moreover, one major concern with this design is the orientation of the interface. Most computer screens are wider than they are high. Figure 4.2

below shows the prototype rendered in a web browser having the 4:3 aspect ratio most commonly used in computer monitors.

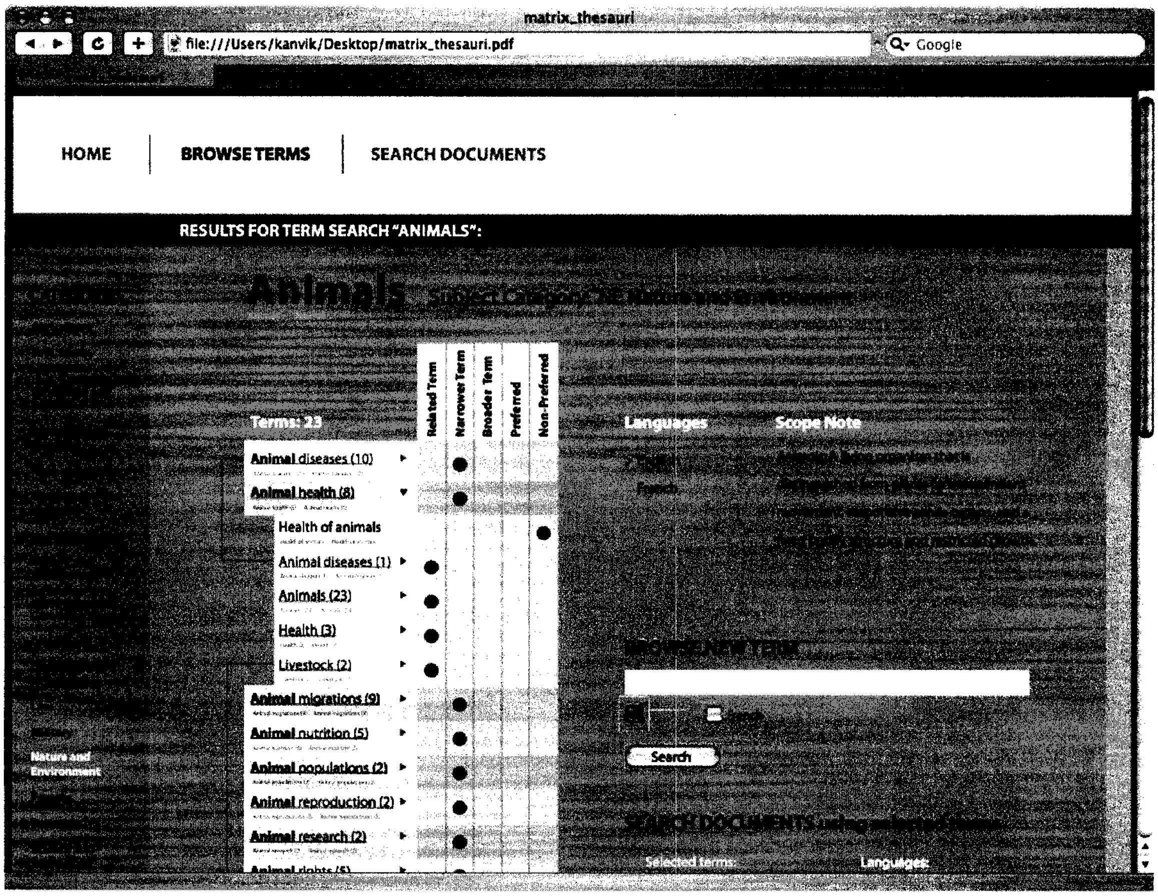

Figure 4.2 – The prototype as it would be rendered in a web browser

Because this interface is significantly longer than it is wide, at any given time a great deal of information is hidden and out of view from the user. The location of the document space in particular is problematic. Any time search results are viewed, the user is required to scroll a great deal to the bottom of the window.

Finding a solution to this problem is not trivial. Significantly reducing white-space to fit more elements into the viewable area would make the interface more cluttered, and doing so may not actually solve the problem. In addition, reducing font sizes would decrease legibility. Rearranging the position of each of the three spaces would still require a great deal of scrolling. One option may be to offer the thesaurus space as an overlay on top of the query space and the document space, allowing the document space to fill the right hand portion of the screen. The user would then be provided with a means for easily switching between the overlay and the remainder of the interface. The implications of this alternative are relatively unknown and would require sketching out a new prototype with a significantly altered layout.

4.6 – Truth

An interface that conforms to the validity claim of *truth* must provide the user with information that is accurate. The system must be bug-free and execute commands as detailed in the system's specification. Documentation must be accurate and current. References to the interface including screen shots must match the implementation.

Because this system is still a conceptual prototype, it has not reached implementation and thus has not been subjected to the development process where systemic errors might be introduced. As such, we cannot say at this time that the system is not prone to error. In addition, the system does not currently have a help system or any other form of documentation available for inspection. However, there are a few items that can be scrutinized to determine the likelihood that the system will provide accurate information to the user.

The first aspect to consider is the system's ability to return the correct documents that are relevant to the query submitted by the user. An information retrieval system can be evaluated using a number of performance measures. While a variety of formulae can be applied, most performance measures take into account two things: *precision* and *recall*. Precision refers to the proportion of retrieved documents that are relevant to the submitted query to the total number of documents that were retrieved. In other words, how many retrieved documents were actually relevant to the search criteria? Recall, on the other hand, refers to the proportion of relevant documents that were retrieved to the total number of relevant documents that were available. In other words, how many documents relevant to the search criteria were actually retrieved by the system? Because both precision and recall are calculated based on relevance, it should be mentioned that in practice relevance is subjective, and is based on what the individual user feels is relevant.

Different performance measures weight these two metrics differently. For example, the F_1 measure weighs precision and recall equivalently, whereas the F_2 measure weighs recall twice as much as precision. In general, a system can be considered to produce accurate results when it has both high precision and high recall. A system that, when evaluated, produces an *F*-measure of 1.0 is considered to be 100% accurate. Such a system not only returns all documents that are relevant to the query, but it also does not return any irrelevant documents. This is of course the ideal and virtually unachievable in reality.

Many information retrieval systems use complex algorithms to evaluate the relevance of a document against a given set of search criteria. These systems are typically performing free-text searches, where every word in the document is considered in determining relevance. For example, search systems on the Web, such as Google™ and Yahoo!®, have developed elaborate ranking algorithms for determining the relevance of web pages against a user-submitted query. These algorithms may also calculate a relevance score based on page popularity and link popularity to determine the order in which the documents are to be placed. Systems that incorporate a controlled vocabulary, on the other hand, make use of a tagged document set to determine relevance. Some algorithmic calculation may be used as well to determine the order in which results are presented to the user, but the controlled vocabulary is the primary resource used by the system to determine which documents are included in the result set.

The primary challenge of information retrieval systems is effectively finding the information the user is looking for. Some systems have good algorithms and others have less effective algorithms. Systems that attempt to rank documents in some sort of order of relevance may work well in some instances, but may not always provide the user with optimal results, because relevance can be subjective. The system may attempt to guess which documents are best, placing these documents at the top of the list, implicitly declaring to the user that these documents are the best, or most relevant documents and likely contain the information required. However, the user may have a completely different idea of what is relevant, and the desired document may actually appear in the middle or perhaps the bottom of the list. Hence, systems that make claims of relevance run the risk of misleading the user into believing that the only documents worth investigating are those ranked within the top five or ten documents in the list of results. If the system makes a relevance claim that turns out to be false, the system has violated the validity claim of truth.

The system presented here dispenses of the need for developing a complicated algorithm through the use of a controlled vocabulary (the thesaurus) to determine relevance. No calculations whatsoever are used to determine which documents are included in the result set and which documents are ignored. If the user adds a thesaurus term to the search query, the system returns all the documents tagged with that term. Furthermore, the interface does not make any claims with respect to relevance. The documents are simply placed in some order selected by the user. This could mean that the documents appear with the newest documents at the top, or simply in alphabetical order by the author's name. Because the system does not attempt to make any guesses on behalf of the user, it does not run the risk of providing misinformation. In this regard, this system conforms to the validity claim of truth simply by virtue of the simplicity of the retrieval process. In essence, conformity to the validity claim of truth can be established by reducing what the system claims to be true, and by only making claims that are completely and verifiably true. In this case, it is the user who determines whether or not a document is relevant, not the system.

There is one caveat, however. The accuracy of this system is highly dependent on the accuracy of the tagging of the document set. For example, if a document mentions livestock feed, but has not been tagged with the term "animal nutrition", the system will miss that document when the user submits a query with the term "animal nutrition". If this occurs, the system as a whole will violate the validity claim of truth. This is not the fault of the interface itself, but a drawback of the architecture of the system, which is susceptible to human error introduced in the tagging process. However, the advantage of using a controlled vocabulary far outweighs the relatively low likelihood that human error will prevent the user from finding the desired information.

4.7 – Truthfulness

In the previous chapter, truthfulness was shown to be the sincerity of an utterance as perceived by the listener. While the difference between truth and truthfulness is subtle, truthfulness considers whether or not the listener is able to believe that the speaker is truly genuine and that the speaker is really being honest in their intentions. Truthfulness is partially dependent on the accuracy (truth) of the utterance, but is reliant a great deal more on the credibility of the delivery of that utterance. In short, does the speaker appear trustworthy? If the user feels in any way as though the system is hiding information, confidence in the accuracy of the information decreases, resulting in a violation of the claim of truthfulness. This can be achieved through adequate feedback to the user and through creating transparency between the system image and the underlying structure of the system.

The prototype shown in figure 4.1 is influenced partly by Bertin's (2001) notion regarding the use of matrices in user interfaces. Bertin postulates that when elements of a user interface are arranged in the form of a matrix, the user is more able to decipher the underlying structure of the system. Recalling the position of Winograd and Flores, designs that are closely aligned with the underlying structure of the system are more effective in communicating to users how they are to successfully operate the system. In terms of Norman's conceptual model (see figure 2.2 on page 22), the user is able to create an accurate model of the system as system image mirrors the actual underlying structure of the system.

Since the users of this system will browse the document collection through the thesaurus, the underlying structure of this multilingual information retrieval system is essentially the thesaurus itself, including all of the relationships between terms. Displaying the thesaurus in the form of a matrix provides the user with a visual overview of the relationships that exist within the thesaurus, how terms are related and which terms are preferred over others. Because the user can quickly ascertain the relationships between terms, it is much easier to navigate and browse through those relationships. As a result, the user is more capable of finding the terms best suited to formulate a search query, and is more efficient and confident in the quality of those terms. As confidence in the quality of queries submitted to the system increases, the user gains greater confidence in accuracy of the returned result set.

It is difficult to evaluate the effectiveness of system feedback given a static prototype. We are unable to determine how the system responds to user requests and how the user is kept informed while the system is processing these requests. One aspect of feedback that we can see is found within the document space of the interface where the resulting set of documents is displayed to the user. Near the bottom of the screen above the space where the documents are listed, we see the text "Retrieved Documents: 6". This space provides an opportunity to inform the user not only how many documents were retrieved, but also *how* they were retrieved. For example, in the case of a search with a Boolean OR, the text could read: "6 documents are tagged with at least one of the following terms: animal diseases, animal health, and animal nutrition". If the search was submitted with a Boolean AND, the text could alternatively read: "6 documents are tagged with each of the following terms: animal diseases, animal health, animal nutrition". This informs the user that the system has acknowledged the user's query, and more specifically which terms the user has submitted to the system. Such feedback also provides descriptive information on how the search was conducted. The user receives textual confirmation of the Boolean operator used in retrieving the documents, and the user is informed that the system is looking at a set of tags associated with each of the documents rather than performing a free text search. This small textual clue provides the user with a great deal of information about how the system works behind the scenes. The more a user is able to understand how the system works, the more confident the user is in the information provided by the system.

In addition to the textual clues described above, it would also be advantageous to provide a mechanism for viewing all the terms with which any given returned document has been tagged. Such a mechanism would provide even more transparency to the user, specifically how the documents are being tagged. This would provide the user with even greater understanding of how the system is structured, inducing even more confidence in the accuracy of the system.

4.8 – Normative Right

An interface that conforms to normative rightness is aware of the larger social concerns of how humans communicate and how they interact with computer systems. When two actors meet for the purpose of exchanging information, they must decide on a set of conventions and use words and semiotic systems that both individuals can agree upon. Designers of user interfaces must be sensitive to the various technical, linguistic and cultural backgrounds of their user base.

The prototype in figure 4.1 shows how this system could be used for searching documents in either English or French. It is safe to assume, then, that users of the system would be proficient in either one or both of these languages. The system has mechanisms throughout the interface for facilitating switching between languages. The user is able to switch to either an English view or a French view of the system, depending on their preferred working language. In the table of thesaurus terms, below each term there is a list of translations in microtext. This is particularly useful for those who are bilingual and might wish to see how the thesaurus is indexing terms in other languages. The document space also provides an option for switching the language for retrieved documents.

Other interface elements, such as check boxes, drop-down pick lists, and other buttons are ubiquitous in all modern graphical user interfaces. As such, users should not have any trouble discerning their meaning, how they are used, and what they do. The terminology used is simple and straightforward. The expressions "related term", "narrower term" and "broader term" should be familiar to those who have never used a thesaurus before.

Given that the target languages in this interface are English and French, it is expected that the users would expect tasks to be ordered sequentially using a left-to-right, top-to-bottom reading sequence. Mullet and Sano (1995) state that hierarchy should be created among groupings "with perceptual prominence corresponding to the intended reading sequence". For the most part, this interface follows this convention. The user begins by browsing the thesaurus on the left, building a search query on the right, and seeing the results at the bottom of the screen. The only exception seems to be the language switch. If the intention is for the user to switch into their preferred operating language before proceeding to browse the thesaurus, build and execute queries, then perhaps this feature should not be placed to the right of the thesaurus space, but somewhere in the top-left portion of the screen.

4.9 – Summary

The design of a quality, usable user interface brings a number of challenges to the development of information retrieval systems. In this chapter, a graphical prototype for a thesaurus-based multilingual information retrieval system is presented. The purpose of the development of this interface was to investigate how to successfully integrate the thesaurus into the process of searching for documents in a multilingual environment. Evaluation of the proposed interface based on Habermas's theory of communicative action has brought to light both positive and negative aspects of the interface.

Assuming an accurately tagged document set, this interface is able to provide the user with accurate results, ensuring conformity to the validity claim of truth. This is a result of using a thesaurus-based controlled vocabulary at the heart of the system, which avoids the necessity for developing complicated algorithms to determine relevance on behalf of the user. By aligning the underlying structure of the system with the user interface, the user is also more confident in the results provided by the system. Aided through the use of a matrix, this approach also considerably enhances the perceived truthfulness of the system.

Overall, the comprehensibility of the system is adequate; however, the orientation of the interface may prevent some important elements from being visible at all times. As the product moves into the implementation stage, it is important that the interface provides some means for accessing documentation and help files. In general, the system conforms to normative rightness, with the exception of the logical ordering of some elements in the display. One significant concern with this interface stems from the way in which the prototype was developed. Neglecting to both identify and consult with a target user base leaves the impression that the designers do not satisfactorily understand the needs of their potential users.

Chapter 5 : Conclusion

The ability to navigate and locate information is becoming increasingly important in our modern world. The ability to overcome language barriers while searching for information is also significant as our society matures into a global community. Despite the many recent technological improvements in our ability to communicate over large distances, language barriers continue to impede the sharing of ideas and information across cultures. Research in the area of information retrieval has produced increasingly advanced systems that allow users to quickly and easily locate the information they require, and multilingual information retrieval systems have also seen significant improvements. These systems give the user the ability to obtain information recorded in multiple languages while operating in and submitting queries using a single language.

While the design of early information retrieval systems neglected serious attention to the role of the end-user, recent awareness of the importance of the user has brought improvements to the usability of such systems. Unfortunately, many interfaces that are designed specifically to work with multiple languages have not seen significant progress in their usability. A prototype of a graphical user interface for a thesaurus-based multilingual information retrieval system has been presented and evaluated according to a proposed theory of human-computer pragmatics, based on Habermas's theories in universal pragmatics and adapted in a manner appropriate for computer interfaces.

Human-computer pragmatics views the user and the designer as having a social relationship, where each individual contributes to a conversation through communicative acts. Through the system image, the designer provides the user with information about what actions can and cannot be performed, where the user can go, what information is relevant to the task at hand, and so on. The user, in turn, responds to the designer by interacting with the device, primarily through the manipulation of symbols on the screen. With a more sophisticated understanding of how we as human beings communicate with one another, designers can create computer systems that are capable of better communication with the user. As communication between the user and the designer improves, the user is less likely to become frustrated, is more confident, and is ultimately more productive.

Habermas's theory of communicative action offers four principles which, when adhered to, allow true communication to occur. The four "validity claims" whereby a speaker may defend their utterances are: comprehensibility, truth, truthfulness, and normative right. First and foremost, the user must understand the interface, including what actions they can perform and the results of those actions. The system must also provide the user with accurate information while instilling within the user a feeling of trust that the information provided is indeed true.

The language, symbols and procedures must be communicated in a way with which both the user and the designer can agree. For designers this means understanding the cultures, practices and vernacular of their users. For users this means making an effort to understand how to operate a software system. This may be as simple as learning how to operate a mouse and a keyboard, or as complex as learning series of commands used to perform more advanced tasks.

Communication between users and designers breaks down as either or both parties employ what Habermas refers to as strategic action, or action that is goal oriented and self-centred. Designers should avoid designs that are not in the best interest of the end-user, such as interfaces that sacrifice functionality in favour of a design that is stylish or trendy. Designers should not add features for the sole purpose of demonstrating their own personal skilfulness, but only if they are useful to the end-user and do not compromise the overall usability of the interface. Finally, project management should avoid making decisions that are strictly financially or otherwise self-motivated at the expense of the end-user, such as prematurely releasing a product, which is bug-laden and unstable. In general, designers should be aware of any egocentric goals that they might have and work to prevent these attitudes from affecting the usability of their systems.

The prototype presented in chapter four was evaluated according to these principles and was found to conform to all four validity claims, although in varying degrees. A number of improvements to the interface were suggested in order to improve conformity to these validity claims, including altering the position of some of the elements on the screen. The architecture of the proposed system was shown to incorporate a number of features that enhance communication. The use of a controlled vocabulary, coupled with an interactive, browsable matrix, gives the user confidence that the system is providing accurate information. A controlled vocabulary, in this case a multilingual thesaurus, provides the user with discrete results when performing queries. The matrix then provides a snapshot of a specific term in the thesaurus, how it is related to other terms and how it is used in the indexing of documents.

The evaluation included in chapter four looks primarily at the communicative features expressed by the designers. Because a communicative event requires more than one individual, the participation of the end-user in the conversation must be considered as well. In order to carry out a complete evaluation of the interface where communicative acts of both parties are examined, thorough user testing must be performed. Ideally, some form of consultation with potential users should take place before initiating the costly implementation process. Without question, a functional prototype should be presented to a group of potential users for feedback before releasing a working version to the general public.

Once a functional system is ready for user testing, additional evaluation can take place of not only the interface itself, but the concept of human-computer pragmatics as well. While the evaluation in chapter 4 consists exclusively of a qualitative exploration of the prototype by the author, additional qualitative criticisms by end-users and quantitative measures through additional user consultation would improve the overall assessment of the application of universal pragmatics to the area of human-computer interaction. A future user study should involve an observational component to gauge how users interact with a working system. Having provided end-users with the opportunity to use a working system, a questionnaire can be distributed to gauge the users' feelings about the interface's ability to communicate to the user. For example, does the interface effectively communicate to the user what actions they are to perform? Does the user trust the accuracy of the system and feel that the results returned are suitable given a submitted query? Is there anything about the interface that is distracting or confusing? A revised design can then be produced, incorporating user feedback collected during the study.

In a period where technology is becoming progressively ubiquitous, communication between individuals, communities and nations is becoming less limited. As technology continues to improve our ability to communicate with individuals throughout the world, it is becoming increasingly important to be able to cross language and cultural barriers and share information on a global scale. As we continue to depend more and more on technology in our communications with others, we must ensure that these devices are indeed enhancing our communications and not amplifying noise in the channel of communication.

Due to our increasing dependency on technology, it is essential for us to be able to communicate effectively with our own creations. Unfortunately, many computer systems and other devices are poor communicators, resulting in less usable systems and more frustrated users. This is a direct result of designers who do not understand how to effectively communicate with their users through a device's interface. As designers understand how human beings communicate with each other, they become more effective at building systems that are successful communicators. The result is a populace of happier, more confident and more productive operators of technology.

Bibliography

Ågerfalk, Pär J., Göran Goldkuhl, and Stefan Cronholm. 1999. Information systems actability engineering - Integrating analysis of business processes and usability requirements. In *Proceedings of the 4th international workshop on the language action perspective (LAP99)*.

Agostini, Alessandra, Giorgio De Michelis, and Maria Antonietta Grasso. 1997. Rethinking CSCW systems: The architecture of MILANO. In *Proceedings of the 1997 fifth European conference on computer supported cooperative work*, edited by J. A. Hughes, W. Prinz, T. Rodden and K. Schmidt. Dordrecht: Kluwer Academic Publishers.

Arnold, Doug. 2003. Why translation is difficult for computers. In *Computers and translation: A translator's guide*, edited by H. L. Somers. Amsterdam; Philadelphia: John Benjamins Pub. Co.

Austin, J. L. 1962. *How to do things with words*. Oxford, England: Oxford University Press.

Benoît, G. 2001. Critical theory as a foundation for pragmatic information systems design. *Information research* 6 (2).

Bertin, J. 2001. Matrix theory of graphics. *Information design journal* 10 (15):5-19.

Bowers, J., and J. Churcher. 1989. Local and global structuring of computer mediated communication: Developing linguistic perspectives on CSCW in Cosmos. *Office: Technology and people* 4 (3):197-227.

Bush, Vannevar. 1945. As we may think. *The Atlantic monthly* 176 (1):101-108.

Caroll, John M. 2000. *Making use: Scenario-based design of human-computer interactions*. Cambridge, MA: MIT Press.

Chowdhury, G. G. 1999. *Introduction to modern information retrieval*. London: Library Association Publishing.

Cline, Andrew R. 2007. *Rhetorica*. http://www.rhetorica.net/canon.htm (accessed July 2007).

Dietz, J.L.G. 1994. Business modeling for business redesign. In *Proceedings of the 27th annual Hawaii international conference on system sciences (HICSS '94)*. Los Alamitos: IEEE Computer Society Press.

Erickson, Tom D. 1990. Working with interface metaphors. In *The art of human computer-interface design*, edited by B. Laurel: Addison-Wesley.

Gibson, Eleanor J., and Arlene S. Walker. 1984. Development of knowledge of visual-tactual affordances of substance. *Child development* 55 (2):453-460.

Gibson, James J. 1979. *The ecological approach to visual perception*. Boston: Houghton Mifflin Company.

Government of Canada Core Subject Thesaurus. 2004. http://en.thesaurus.gc.ca/intro_e.html (accessed April 2007).

Grefenstette, Gregory. 1998. The problem of cross-language information retrieval. In *Cross-language information retrieval*, edited by G. Grefenstette: Kluwer Academic Publishers.

Habermas, J. 1998. What is universal pragmatics? In *On the pragmatics of communication*, edited by M. Cooke. Cambridge, Massachusetts: The MIT Press.

Harris, Robert A. 2005. *A handbook of rhetorical devices* http://www.virtualsalt.com/rhetoric.htm (accessed July 2007).

Hearst, M. 1999. User interfaces and visualization. In *Modern information retrieval*, edited by R. Baeza-Yates and B. Ribeiro-Neto. Reading, MA: Addison Wesley.

Hearst, M., and C. Karadi. 1997. Cat-a-Cone: An interactive interface for specifying searches and viewing retrieval results using a large category hierarchy. In *Proceedings of the 20th annual international ACM/SIGIR conference on research and development in information retrieval (SIGIR `97)*. Philadelphia, PA.

History magazine. 2001. http://www.history-magazine.com/libraries.html (accessed July 2004).

Hochheiser, Harry, and Ben Shneiderman. 2001. Universal usability statements: Marking the trail for all users. *ACM interactions* 8 (2):16-18.

Hull, D., and G. Grefenstette. 1996. Querying across languages: A dictionary-based approach to multilingual information retrieval. In *Proceedings of the 19th international conference on research and development in information retrieval.*

Hutchins, W. John. 2001. Machine translation over fifty years. *Histoire, epistemologie, language* 22 (1):7-31.

Instone, Keith. 2004. *UniversalUsability.org*, (October 2004) http://universalusability.org/node/38 (accessed July 2007).

ISO-2788. 1986. *Guidelines for the establishment and development of monolingual thesauri*. 2nd ed: International Organization for Standardization.

ISO-5964. 1985. *Guidelines for the establishment and development of multilingual thesauri*: International Organization for Standardization.

Jones, G.O., and A. Basden. 2002. How Habermas' action types can influence KBS design. *Research and development in expert systems* 6 (1).

Jordan, P.W. 2002. *Designing pleasurable products: An introduction to the new human factors*. London, UK: Taylor and Francis.

Jorna, K., and S. Davies. 2001. Multilingual thesauri for the modern world - no ideal solution? *Journal of documentation* 57 (2):284-295.

Jurafsky, Daniel, and James H. Martin. 2000. *Speech and language processing*: Pearson Education.

Korn, F., and B. Shneiderman. 1995. Navigating terminology hierarchies to access a digital library of medical images: University of Maryland.

Lin, X. 1999. Visual MeSH. In *Proceedings of 22nd annual international ACM/SIGIR conference on research and development in information retrieval (SIGIR '99)*, edited by M. Hearst, F. Gey and R. Tong. New York: ACM.

Ljungbert, J., and P. Holm. 1996. Speech acts on trial. *Scandinavian journal of information systems Scandinavian journal of information systems* 8 (1).

Lyytinen, Kalle, Erkki Lehtinen, and Esa Auramäki. 1987. SAMPO: A speech-act based office modelling approach. *ACM SIGOIS Bulletin* 8 (4):11-23.

Marcus, Aaron, and Emilie West Gould. 2000. Crosscurrents: Cultural dimensions and global Web user-interface design. *ACM interactions* 7 (4):32-46.

McMath, C.F., C.S. Tamaru, and R. Rada. 1989. A graphical thesaurus-based information retrieval system. *International journal of man-machine studies* 31:121-147.

Medina-Mora, R., T. Winograd, R. Flores, and C.F. Flores. 1992. The action workflow approach to workflow management technology. In *Proceedings of the 4th conference on computer-supported cooperative work*, edited by J. Turner and R. Kraut. Toronto, Canada.

Mullet, K., and D. Sano. 1995. *Designing visual interfaces*: Prentice Hall.

Newall, Paul. 2005. *Rhetoric*. http://www.galilean-library.org/int21.html (accessed June 2007).

Nielsen, Jakob. 1993. *Usability engineering*. San Diego, CA: Academic Press, Inc.

Norman, Donald A. 1988. *The design of everyday things*. New York: Basic books.

———. 2004. *Design as communication*. http://www.jnd.org/dn.mss/design_as_communicat.html (accessed May 2007).

———. 2004. *Emotional design: Why we love (or hate) everyday things*. New York: Basic Books.

———. 2005. Human-centered design considered harmful. *ACM interactions* 12 (4):14-19.

Raley, Rita. 2003. Machine translation and global English. *Yale journal of criticism* 16 (2):291-313.

Rimmer, J. 2004. Improving software environments through usability and interaction design. *Journal of audiovisual media and medicine* 27 (1):6-10.

Schoop, M. 1998. A language-action perspective on cooperative documentation systems - Habermas and Searle in hospital. In *Proceedings of the 3rd international workshop on the language action perspective on communication modelling*: Department of Informatics, Jonkoping International Business School.

Schoop, Mareike. 2001. An introduction to the language-action perspective. *SIGGROUP bulletin* 22 (2):3--8.

Searle, John R. 1969. *Speech acts: An essay in the philosophy of language*. Cambridge: Cambridge University Press.

———. 1979. *Expression and meaning: Studies in the theory of speech acts*. Cambridge: Cambridge University Press.

Shalizi, Cosma. 1995. *Fools can be so ingenious.* http://cscs.umich.edu/~crshalizi/reviews/everyday-things/ (accessed December 2006).

Shannon, C.E. 1948. A mathematical theory of communication. *The Bell System technical journal* 27:379-423, 623-656.

Shiri, A., C. Revie, and G. Chowdhury. 2002. Thesaurus-enhanced search interfaces. *Journal of information science* 28 (2):111-122.

Shneiderman, Ben. 1997. *Designing the user interface: Strategies for effective human-computer interaction.* Reading, MA: Addison-Wesley.

———. 2000. Universal usability. *Communications of the ACM* 43 (5):84-91.

Silva Rhetoricae. 2007. http://rhetoric.byu.edu (accessed July 2007).

Sjöström, J., and P. Ågerfalk. 2004. Analysis of communicative features of user interfaces. In *Proceedings of the 2nd International Conference on Action in Language, Organizations and Information Systems (ALOIS-2004).* Linköping University.

Sjöström, J., and G. Goldkuhl. 2002. Information systems as instruments for communication - Refining the actability concept. In *Proceedings of the 5th international workshop on organizational semiotics (OS 2002).* Delft, The Netherlands.

Stubblefield, William A. 1998. Patterns of change in design metaphor: A case study. In *Proceedings of the SIGCHI conference on human factors in computing systems (CHI '98).* New York, NY, USA: ACM Press/Addison-Wesley Publishing Co.

Te'eni, Dov. 2006. The language-action perspective as a basis for communication support systems. *Communications of the ACM* 49 (5):65--70.

Vronay, David. 2006. *The UI design minefleld - er... flower field??,* (September 2006) http://shellrevealed.com/blogs/shellblog/archive/2006/09/26/The-UI-design-minefield-_2D00_-er_2E002E002E00_-flower-field_3F003F00_.aspx (accessed June 2007).

Whitelock, P.J., M. McGee, B.J. Chandler, N. Holden, and H.J. Horsfall. 1986. Strategies for interactive machine translation: The experience and implications of the UMIST Japanese project. In *Proceedings of the 11th international conference on computational linguistics (COLING-86).*

Wijnia, Elmine. 2004. Understanding weblogs: A communicative perspective. In *BlogTalks 2.0: the European conference on weblogs*, edited by T. N. Burg.

Wilks, Yorick, Kenneth W. Church, Sergei Nirenburg, Eduard H. Hovy, and Craig A. Knobllock. 1996. Statistical versus knowledge-based machine translation. *IEEE intelligent systems* 11 (2):12-18.

Winograd, T. 1988. Computer-supported cooperative work: A book of readings, edited by I. Greif. San Mateo: Morgan Kaufman.

Winograd, T., and F. Flores. 1986. *Understanding computers and cognition: A new foundation for design*. New Jersey: Ablex Publishing Corporation.