Using a Common Accessibility Profile to Improve Accessibility

A Thesis Submitted to the
College of Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
David W. Fourney

Permission to Use

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

Abstract

People have diffculties using computers. Some have more diffculties than others. There is a need for guidance in how to evaluate and improve the accessibility of systems for users. Since different users have considerably different accessibility needs, accessibility is a very complex issue.

ISO 9241-171 defines accessibility as the "usability of a product, service, environment or facility by people with the widest range of capabilities." While this definition can help manufacturers make their products more accessible to more people, it does not ensure that a given product is accessible to a particular individual.

A reference model is presented to act as a theoretical foundation. This Universal Access Reference Model (UARM) focuses on the accessibility of the interaction between users and systems, and provides a mechanism to share knowledge and abilities between users and systems. The UARM also suggests the role assistive technologies (ATs) can play in this interaction. The Common Accessibility Profile (CAP), which is based on the UARM, can be used to describe accessibility.

The CAP is a framework for identifying the accessibility issues of individual users with particular systems configurations. It profiles the capabilities of systems and users to communicate. The CAP can also profile environmental interference to this communication and the use of ATs to transform communication abilities. The CAP model can be extended as further general or domain specific requirements are standardized.

The CAP provides a model that can be used to structure various specifications in a manner that, in the future, will allow computational combination and comparison of profiles.

Recognizing its potential impact, the CAP is now being standardized by the User Interface subcommittee the International Organization for Standardization and the International Electrotechnical Commission.

Acknowledgements

My graduate education, including this work, would not have been possible without the support and caring of several people.

I would like to thank my wife, Colleen for her love and patience. At the start of this journey, I had been advised that graduate school is hard on a marriage. Colleen's dedication and support helped me to focus on my work in the most stressful times.

I would like to thank Dr. Jim Carter for believing in me. From championing my admission to the College of Graduate Studies to advocating for my eventual admission into the Master's program, I would have given up the fight if it had not been for his unwavering support. Further, I would like to thank Jim for his patience, guidance and support over the past five years which has culminated in this work.

Thank you to the other members of my Committee. Thanks to both Dr. John Cooke and Dr. Jean-Paul Tremblay for taking time out of their retirement to support me. Thanks also to Dr. Gregg Vanderheiden of the University of Wisconsin-Madison for agreeing to examine this Thesis, and to Dr. Grant Cheston, a big thank you for agreeing to chair the examination on short notice.

Finally, I want to thank the members of ISO/TC 159/SC 4/WG 5 and ISO/JTC 1/SC 35 for their contributions to this work. The ongoing development of ISO/TS 16071 and ISO 9241-171 during the period of my studies lead to several ideas that eventually made their way into this work. The interest in my work from ISO/JTC 1/SC 35 boosted my confidence and my desire to continue the effort.

In memory of my grandmother, Ella.

She was a mother, teacher, author, and community leader
who never allowed her own disabilities to limit her dreams.
While she will never benefit from this work,
she was a source of inspiration for it.

Contents

List of Tables

List of Figures

List of Abbreviations

ACCLIP	IMS Accessibility and Learner Information Profile
ACCMD	IMS Accessibility Metadata
AT	Assistive Technology
ANSI	American National Standards Institute
CAP	Common Accessibility Profile
CAP_{AT}	Assistive Technology CAP Specification
CAP_{CF}	Component Feature CAP Specification
CAP_{ENV}	Environment CAP Specification
CAP_{IC}	Interacting Component CAP Specification
CAP_{IR}	Input Receptor CAP Specification
CAP_{OT}	Output Transmitter CAP Specification
CAP_{PF}	Processing Function CAP Specification
CAP_{SYS}	System CAP Specification
CAP_{USE}	User CAP Specification
CD	ISO Committee Draft
CF	Component Feature
CIF	Common Industry Format for Usability Test Reports
DPI	Disabled Peoples' International
FCD	ISO Final Committee Draft
FLOSS	Free/Libre/Open-Source Software
IC	Interacting Component
IM	Instant Messaging
IR	Input Receptor
ICF	International Classification of Functioning, Disability, and Health
IEC	International Electrotechnical Commission
IMS	Instructional Management Systems Global Learning Consortium, Inc.
ISO	International Organisation for Standardisation
ISO/TS	ISO Technical Specification
LIP	IMS Accessibility for Learner Information Package
OT	Output Transmitter
PAN	Personal Area Network
PC	Personal Computer
PDA	Personal Data Assistant
PF	Processing Function
RA	Registry Authority
SI	Système International
UARM	Universal Access Reference Model
UNS	User Needs Summary
URL	Uniform Resource Locator
USB	Universal Serial Bus
W3C	World Wide Web Consortium
WHO	World Health Organization
XML	Extensible Markup Language

List of Listings

Chapter 1

Introduction

People can have difficulties using computers. These people can be of any age, ability, or background. Some people have more difficulty than others.

1.1 Improving Accessibility of Computing Systems

Numerous studies have identified different usability issues. One study comparing the usability experience of several web sites for both younger and older adults found that overall usability was slightly more than twice as good for non-seniors as it was for seniors (Nielsen, 2002). A survey of the role of computer-based Assistive Technology (AT) in Canadian post-secondary environments found that, among post-secondary students with disabilities, although 95% of the 800 respondents use a computer, only a quarter of respondents currently use an AT with their computer and half report the need for an AT (Fichten, Asuncion, & Barile, 2001). Research experience with children suggests that product usability is closely related to children's enjoyment (Hanna, Risden, Czerwinski, & Alexander, 1998).

While some will blame the people who have difficulty, others will blame the computer. What is important is to remove the difficulty without focusing on the blame. Designing universally usable systems is one way to do that.

1.1.1 Usability and Accessibility

Different people have different impressions about usability and accessibility. The concepts of *usability* and *accessibility* are closely related to each other. They are also separate and distinct.

International Organisation for Standardisation (ISO) 9241-11 is an international standard that describes how to identify the information required when specifying or evaluating usability in terms of measures of user performance and satisfaction. This standard provides guidance on how to describe the context of use of the product and the measures of usability in an explicit way. It also includes an explanation of how the usability of a product can be specified and evaluated as part of a quality system (International Organization for Standardization [ISO], 1998a).

The definition of usability in ISO 9241-11 describes a composite of effectiveness, efficiency, and satisfaction with which specified users achieve specified goals in particular environments:

> **usability**
> the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use (ISO, 1998a)

The three terms *effectiveness*, *efficiency*, and *satisfaction* refer respectively to how well the system does the job for the user, how easily the job is done for the user, and how users feel about the process overall. In this context, the notions of effectiveness and efficiency are highly coupled. Thus, usability focuses on specific users and may ignore or exclude other users.

Usability problems impact all users equally, regardless of ability. A person with a disability is not disadvantaged to a greater extent by usability issues than a person without a disability (Thatcher et al., 2002). A usability issue becomes an accessibility issue when it has a disproportionate affect on people with disabilities (Vanderheiden, 2007).

When accessibility is separated from usability, it is often perceived as a "special case" of usability where special designs are needed to fit the requirements of users with specific types of disability. This view suggests that specific users with specific disabilities should be required to purchase a significantly more expensive version of common consumer products which have been specially designed to meet their needs. This is not practical because each user group is so small that no product can be provided at reasonable cost to the consumer (or producer) (Vanderheiden, 1990). In addition, this approach does not meet the needs of the majority of consumers with disabilities because it excludes the widest possible range of users — specific designs for specific groups of users with disabilities by definition miss other specific groups of users with disabilities.

Sometimes this "special case" perspective can lead designers to believe that everything needs to be designed to be used by everyone. This is based on a misperception that everything needs to be designed so that it is accessible to every possible person with a disability. It may be "impractical, if not impossible, to design everything so that it is accessible by everyone regardless of their limitations" (Vanderheiden, 1990) and it may be "unreasonable to design everything so that it can be used by everyone" (Vanderheiden, 1990).

The danger with the "special case" view of accessibility is:

- A product could be "usable" without being "accessible." That is, when designers view accessible design as "outside of" usable design, they have "permission" to design an inaccessible product.

- A product could be "accessible" without being "usable." This occurs when a designer of an "accessible" product does not account for the usability requirements of the specific population for which the product is designed.

However, many people believe that there should be no distinction between *usability* and *accessibility* — "Accessibility is good usability." (Moulton, Huyler, Hertz, & Levenson, 2002) — and thus the term "usability" should encompass the meaning of both terms. According to ISO Technical Specification (ISO/TS) 16071[1], which provides guidelines and recommendations for the design of systems and software that will enable users with disabilities greater accessibility to computer systems, accessibility is defined as:

> **accessibility**
> usability of a product, service, environment or facility by people with the widest range of capabilities (International Organization for Standardization [ISO], 2003b)

Accessibility focuses on the widest range of users recognizing that different users have different needs and may require different accommodations. It is reasoned that, if usability is about producing products and systems that are easy to use and perform the function for which they were designed, then accessible design is about producing products and systems that are usable by all persons regardless of (dis)ability. While usability focuses on a society of "average" users to the exclusion of those who do not fit the "average", accessibility focuses on single individual users with the goal of producing products that are usable by everyone. This definition suggests that the solution lies somewhere between the extremes discussed above: not designing to exclude users with disabilities, not designing specific products for specific users with disabilities, but designing for the widest possible range of users regardless of disability.

This definition suggests that it is possible to design everything for everyone — or to at least get as close as possible. However, such a goal requires designers to consider the needs of all users, including users with disabilities, from the beginning.

For the purpose of this Thesis, the ISO/TS 16071 definition of accessibility will be used.

1.1.2 Creating Accessibility

There are three views of how to create accessibility. Each view has a different focus: the user, the product, or the interaction between the user and the product (Carter & Fourney, 2004b).

Focus on the *User*: This view forces the user to work around the usability of the computing agent and corresponds to bio-medical and functional-rehabilitative models of disability; that is, that a disability is an "illness" to be "treated" or "fixed." The user seems to bear the "blame" for any problems. In this view, the "barrier" is perceived to somehow be an attribute of the user not the system or environment (e.g., Vitense, Jacko, & Emery, 2002 refer to, "individuals with barriers limiting one or more channels of perception, such as a visual impairment").

[1] In 2007, ISO/TS 16071 will be succeeded by ISO 9241-171 (International Organization for Standardization [ISO], 2006f).

In this view, meeting the needs of users and universal access is often seen as a burden rather than as a design goal. For example, International Organisation for Standardisation (ISO) and International Electrotechnical Commission (IEC) Guide 71, which describes a set of guidelines used within ISO to ensure the needs of older persons and people with disabilities are addressed when developing standards, states that accessibility involves, "…extending standard design to people with some type of performance limitation." (2001b). ISO/IEC Guide 71 places the focus of "the problem" squarely on the user. This view has the distinct disadvantage of blaming the users for their limitations (Hong Kong Equal Opportunities Commission, 1999).

Focus on the *Product*: This view is focused on the usability of the computing agent and away from the user. It forces the computing agent to resolve usability problems. However, it continues to see human-computer interaction as a two-party affair and assigns blame for any problems on the computing agent. For example, the ISO/TS 16071 definition of "accessibility" quoted above has the advantage of focusing the "problem" of accessibility on the usability of the computing agent and away from the user, however, it continues to see human-computer interaction as a two-party affair and still assigns blame to one of them.

Focus on the *Interaction* between the user and the product: The problem of accessibility is neither the "fault" of the user nor the system. The focus is on, "…removing barriers that prevent people … from participating in substantial life activities … ." (Bergman & Johnson, 1995). There is a third party involved, the "barrier" — something which handicaps the interaction between the user and the system. This interaction can be described as a negotiation among parties where each party brings their own "terms" into the negotiation and a working partnership is sought.

Attributing "blame" to one side or the other is undesirable. Parties can only communicate if they have some shared context, which may include various types of knowledge and abilities. Anything missing in the shared context of the interaction will handicap the communication between the user or the product (Carter & Fourney, 2004b). Focusing on the *interaction* between the user and the product means that adaptations need not be one-sided.

In the world of computing there are two different kinds of accessibility: physical accessibility (i.e., the "stuff outside the box," such as the keyboard or mouse) and "logical accessibility" (i.e., software which does not require any further adaptation). A focus on the interaction between the user and the product encompasses both of these views.

1.1.3 (Dis)Abilities

To have successful interactions, there needs to be a suitable matching of user and system capabilities. Many people focus on disabilities rather than abilities because they assume that a disability leads to failure. When used in a personal sense (i.e., "disabled person"), the word "disabled" negates a person, and largely dismisses any abilities the person may posses. "Disability" implies that a person is not able to do some things or anything (Baldwin, 2000).

Most definitions of disability touch on only one aspect of disability and not the full range of issues that contribute to disability. A review of contemporary definitions of disability suggests definitions of disability reflect four paradigms (summarized in Table 1.1) which fall mainly into one of two categories of focus: the Individual or the Society (Rioux, 1997).

Table 1.1: Categories of Definitions of "Disability" (adapted from Rioux, 1997)

Category I: Individual as unit of analysis		**Category II: Society as unit of analysis**	
Biological or Medical model	**Functional or Rehabilitation model**	**Environmental model**	**Human rights model**
Emphasis on attributes in the individual.	Emphasis on promoting or restoring fuller functioning in the individual.	Attention directed to ecological barriers: social, economic, political, institutional and legal, which can result in disability.	Focus is on the rights to which all people, including people with disabilities, are entitled.

Definitions that focus on the individual as the unit of analysis tend to centre on either a bio-medical model with emphasis on the attributes of the individual, or on a functional-rehabilitative model with emphasis on the promotion or restoration of fuller functioning in the individual. Such definitions frame disability with reference to the individual and in terms of individual deficits. Most of the models in this category assume a "norm," below which a person should fall if she or he is identified as a person with a disability. Medical definitions have the view that the etiology of disability resides in the individual as a consequence of events such as disease, accident, war, genetic structure, birth trauma, or other acute causes and that by identifying the aetiology involved, disability can be "treated", "cured", or "prevented". The functional-rehabilitative approach has the view that emphasizes the actions or activities that an individual can perform (with or without assistance) as a result of a bio-medical condition (Hong Kong Equal Opportunities Commission, 1999).

Definitions that focus on the society as the unit of analysis tend to centre either on an environmental model with attention directed to ecological barriers (social, economic, political, institutional and legal) which can result in disability, or a human rights model with emphasis on the rights to which all people, including those with disabilities, are entitled. Such definitions focus not on the individual but on the social, economic, political, institutional and legal conditions that can result in disability. Environmental definitions recognize that a physical, sensory or intellectual impairment will not limit many individuals as much as being denied an education, the right to employment, or the right to marry and have a family of their own. Human rights approaches to disability are premised on the recognition of a set of fundamental rights to which all people are entitled regardless of one's individual characteristics, consequently reducing the need for a specific definition of disability (Hong Kong Equal Opportunities Commission, 1999).

While a medical model view of disability ignores the imperfection and deficiencies of the environment (United Nations Commission for Social Development on Disability, 1993), definitions with a focus on society recognize that, "disability has too long been viewed as a problem of the individual and not the relationship between an individual and his/her environment" (Disabled Peoples' International [DPI], 1993). Thus, unlike definitions that focus on an individual's deficits, definitions that focus on society require that the structural conditions in a society that result in disability be addressed and ameliorated.

The International Classification of Functioning, Disability, and Health (ICF) presents a third model, one that attempts to combine what is true of both the medical and social approaches without reducing the entire notion of disability into one or the other's aspects. This "biopsychosocial model" views disability and functioning as outcomes of interactions between health conditions and contextual factors each of which impact how disability is experienced by the individual. Contextual factors include both environmental factors (e.g., social attitudes, legal structures, built or natural environment, climate, etc.) and personal factors (e.g., personal attitude, gender, age, education, past behaviour, etc.). Thus, in the ICF model, disability involves dysfunction at one or more human functioning levels: physical impairment, activity limitations, and participation restrictions (World Health Organization [WHO], 2002).

On their own, although both are partially valid, neither a medical nor social focus fully captures the complexity of disability. Their combination, as suggested by the ICF, does not fully grasp the notion either. The reality is that, "Both the causes and consequences of disability vary throughout the world." (United Nations Commission for Social Development on Disability, 1993). Table 1.1 shows little agreement over the definition of disability because issues such as culture and economics changes the view of what constitutes a "disability" and whether or not a person has a disability from one society to another, in essence, " . . . disability is a social construct." (Kaplan, n.d.)

> Most people believe they know what is and is not a disability. If you imagine 'the disabled' at one end of a spectrum and people who are extremely physically and mentally capable at the other, the distinction appears to be clear. (Kaplan, n.d.)

Just because one is perceived to have a disability in one society does not necessarily mean one is also perceived to have a disability (or even the *same* disability) in another. For example, the island of Martha's Vineyard once had a strain of hereditary deafness such that a large number of residents were born deaf. As a result, most people on the island knew at least one person who was deaf and a large number of islanders were fluent in the local Sign Language. At a time when mainland United States did not allow deaf persons to hold office, much less vote, there was little social differentiation based on hearing status on Martha's Vineyard (Groce, 1985).

Typical definitions of disability refer to an impairment that results in loss of function in one or more major life activities (e.g., walking, seeing, hearing, learning, etc.) The implication of such definitions is that what limits people with disabilities is the inability to see, walk, hear, and so on. While that has obvious validity, the evidence is clear that people with disabilities can live full, productive lives.

Disability is both a problem at the level of one's body as well as a complex social phenomenon. Underestimating the innate abilities of people with disabilities is rooted in archaic notions from a time when it was difficult for people with disabilities to participate in their community. Underestimating people with disabilities results in lowered expectations as to what people can achieve, stereotypes, self-fulfilling prophecies, and attaches a stigma to having a disability (Goodwin, 1997).

Disabilities have to be contextualized. As the World Health Organization (WHO) points out, "Disability is always an interaction between features of the person and features of the overall context in which the person lives, . . . " (WHO, 2002). If in a context you do not need to use a certain ability, then any impairments to that ability are irrelevant. What is relevant are the *abilities* an individual brings into the context.

Before 1980 the word "disabled" was used to label minority groups who had conditions that made them physically, intellectually, or mentally different. In the 1980's the term "people with disabilities" was introduced as social recognition of this minority group as people living in the community increased. Toward the end of the 1990's, changes in technologies pushed the term "disability" out the door. Words such as "ability", "inclusion", and "normalization" have forced a reconsideration of "disabled". This created a new awareness, focusing on the person's *abilities* first, rather than the disability (Baldwin, 2000).

The work of Benjamin Bloom and his colleagues provides a list of abilities to draw on. They identify three domains of educational activities: cognitive, affective, and psychomotor. The cognitive domain has six different major categories and involves knowledge and the development of intellectual skills. The affective domain has five major categories and includes the manner in which we deal with things emotionally, such as feelings, values, appreciation, enthusiasms, motivations, and attitudes. The psychomotor domain has seven major categories and includes physical movement, co-ordination, and use of the motor-skill areas. Development of these skills requires practice and is measured in terms of speed, precision, distance, procedures, or techniques in execution. Each major category of skill can be thought of as a degree of difficulty — one must be mastered before the next can take place (Bloom et al., 1956).

> ["Disability"] is a label of liability. "Ability" is a word of action and asset in the minds and eyes of individuals (Baldwin, 2000).

Instead of focusing on the obvious weaknesses, a focus on abilities recognizes the strengths in an individual (West, 1999) and the opportunities flexibility can create (Gibilisco, 2003). In essence, the idea that there is, "more than one way to skin a cat" (Clemens, 1889) applies.

1.1.4 Handicaps

Historically, the terms "disability" and "handicap" have been misused interchangeably. A handicap is not a disability. A handicap is not caused by a disability. The source of a handicap is the interaction between a person and some other object (including another person). Although the term "handicap" is often associated with persons with disabilities (United Nations Commission for Social Development on Disability, 1993), persons without disabilities also experience handicaps.

The constitution of Disabled Peoples' International (DPI) defines "handicap" as:

> . . . the loss or limitation of opportunities to take part in the normal life of the community on an equal level with others due [to] physical or social barriers (DPI, 1993)

A badly designed system also handicaps users (Thimbleby, 1995). As computers become part of a user's environment, the interaction between users and computers can be seen as a source of handicap; "human-computer interaction is an impoverished affair." (Taubes, 2000). The keyboard and mouse, our primary points of contact, are limiting single mode interactions for use by a species accustomed to multimodal social interaction. In essence, the computer's own shortcomings often create the barriers users, with or without disabilities, experience (Carter & Fourney, 2004b).

For several historical reasons, in the English language, the term "handicap" has been associated with a variety of negative political and emotional connotations. For this reason, the term "barrier" has often been used instead. Unfortunately, without qualification (e.g., the type of barrier), this English term does not fully communicate both the environmental and social impacts the word "handicap" does.

Handicaps are anything that may interfere with the accessibility of interactions. A handicap may have one or many sources among the system, user, interaction, and/or environment. From a universal access perspective, reducing any handicaps to an interaction is far more important than attributing blame (Carter & Fourney, 2004b).

A handicap can provide different levels of interference at different times. On the one hand, a handicap may provide full, partial, or no interference. Full interference means that no interaction between the user and the system can occur and all possible means of interaction are blocked. Partial interference means that only a portion of the many ways the user and system can interact are available. If the number of ways the user and system can interact are so limited that there are no compatible ways for them to interact, a partial interference can still be seen as a full interference. If there is no interference at all, full interaction can occur.

On the other hand, a handicap may be temporary, progressive, or permanent. This temporal dimension to a handicap means, across time, the level of interference in the handicap of the interaction can change as the situation changes. Partial interferences may become full interferences or no interferences. Similarly, a given level of interference may never change (Carter & Fourney, 2004b).

This Thesis will focus on handicaps to interactions rather than disabilities of users.

1.1.5 Focusing on Abilities to Minimize Handicaps

If handicaps interfere with the accessibility of interactions and interactions occur where parties have some shared context, there is a need to find a way to focus on abilities that are shared between the user and the system. The context may include various types of knowledge and abilities. The effect of such a focus would be to minimize handicaps to interactions and maximize accessibility. To accomplish this effect, measurement needs to occur. Such measurement systems do not exist.

1.2 Research Objectives

Since computer users need standardized ways of identifying their access needs to systems and of identifying systems (and their components) that can meet their access needs, there is a need for guidance in how to evaluate and improve accessibility of systems for users.

The notion of "evaluation" suggests the need for descriptors which can be applied to objectively measure accessibility. The notion of "improvement" suggests the need for metrics based on sound principles (e.g., Basili & Rombach, 1988) which can detect change and algorithms to drive and interpret them.

This Thesis will develop a standardized way to describe the needs and capabilities of users and systems.

1.3 Thesis Organization

This Thesis is organized into the following chapters:

- Chapter 1 provides introductory material, motivation and research objectives for the Thesis.
- Chapter 2 discusses accessibility, develops an approach to modelling accessibility (i.e., the Universal Access Reference Model (UARM)), and introduces the Common Accessibility Profile (CAP) as a means to record specifications of the model.
- Chapter 3 defines the structure and contents of information within a CAP that can be used to describe users, systems, assistive technologies, and environments.
- Chapter 4 defines operators on individual pieces and groups of information within a CAP.
- Chapter 5 develops examples of the specification of CAPs for users, systems, and environments.

- Chapter 6 discusses validation of the CAP.
- Chapter 7 discusses the results and contributions of this Thesis and suggests future work.

1.3.1 Defining Common Accessibility Profile Structure and Contents

Chapter 3 begins by determining what is needed by the CAP both in terms of structure and specification. These needs are based on the description of the CAP outlined in Chapter 2. Over several iterations, a structure for the CAP was developed. This structure was specified using tables.

Candidate codings were gathered to fit the identified needs. These codings were initially derived from various resources including Chapter 2, various International Standards, and other applicable research. Where an appropriate candidate coding could not be determined, then a coding was created to fit the identified need. These codings were refined over several iterations until a generalized specification format could be achieved.

Analysis from later chapters of this Thesis was then used to correct and further develop this specification.

1.3.2 Qualifying Information in Relationships within CAPs

Chapter 4 provides the ability to qualify information and specify the relationships between different pieces of information within a single CAP. For example, a specific system could be described as supporting more than one language. It is necessary to qualify a CAP to determine how OR'ing or AND'ing these language capabilities is understood.

This chapter introduces:

- unary operators {SHALL, MAY, NOT} that qualify individual pieces of information, and
- binary operators {AND, OR, XOR} that specify relationships between pieces of information.

1.3.3 Developing Example CAPs

Chapter 5 demonstrates the feasibility of developing meaningful CAPs. These examples fulfill two goals:

1. Evaluating the usability of the CAP approach.
2. Evaluating the completeness and usability of the structure and specification from Chapter 3 and the qualifications and relationships from Chapter 4.

This chapter considered the various examples suggested throughout Chapters 1 and 2. Candidate examples were considered for completeness and applicability to user, system, and environment. Since the number of candidate CAPs was insufficient for all categories, existing examples were modified and new examples were developed.

1.3.4 Validating the CAP

Chapter 6 describes an approach to validation that goes beyond the ability to generate CAPs that is shown in Chapter 5.

This approach involved validation by a large international group of experts who formally reviewed and approved the CAP. Validation also considered the appropriateness of the metrics involved in the CAP by considering their theoretical and practical applicability.

1.3.5 Additional Materials

Technical material is provided in the Appendices as needed to allow the main thesis to present a solution in a manner that can be understood by a general audience (i.e., intended CAP users). A glossary of terms is provided in Appendix A.

Chapter 2

Background

This chapter describes a model-based approach to evaluating and improving accessibility.

Section 2.1 begins with a discussion on standards-based approaches to modelling usability and introduces the UARM. Since handicaps occur within the interactions of users and systems, a model based on interactions is necessary to describe accessibility.

Section 2.2 builds on the introduction of the UARM by describing the components of the model in detail. The model contains five components: users, systems, interactions, environment, and context.

Section 2.3 introduces multi-system models of interaction. Up to this point, the chapter assumes one system interacting with one user. Multi-system models illustrate additional systems, such as Assistive Technologies (ATs), interacting with the user. An AT is a program or device that can be added to a system to make it more accessible. The UARM is expanded to encompass a multi-system model and additional components are discussed in detail.

Finally, Section 2.4 introduces the CAP. The CAP describes user-system and user-AT-system accessibility across all users and systems and is based on interactions, which may involve multiple channels. The CAP provides a framework of descriptors which can be applied to objectively measure usability.

2.1 Modelling Usability

It seems that everyone talks about usability, but it is unclear how to achieve it. Individual approaches have been difficult to generalize outside of the context in which they were developed. There is a need for a repeatable method to achieve usability which can be applied across situations. This method should be based on a model that can be used to evaluate usability both for groups of users and for individual users. This will ensure not only general usability but also accessibility for individuals. Since handicaps occur within the interactions of individual users and systems, a model based on interactions is necessary to describe accessibility.

This section introduces the model of usability used by this Thesis. It discusses the standards-based model of usability and user-centred development and introduces the UARM. The UARM is both a response to the need for models to support system accessibility evaluation and improvement, and a standards-based user-centred approach.

2.1.1 Usability and User-Centred Development

Usability of a product is directly related to its users, their tasks, and the context in which their tasks are performed.

Such differences as the user's experience, skills, and abilities may influence usability. For example, expertise can change how a user prefers to interact with a product. Users who are unfamiliar with a software application or only use it occasionally may be most comfortable using graphical menus. Users very familiar with the same application (so-called "power users"), may be more comfortable using keyboard shortcuts (Carter, 2004).

Different products within the same domain may have differences in usability only by being better for certain tasks (Carter, 2004). For example, Word-processor A may be more usable when inserting and editing page headings than Word-processor B. However, Word-processor B may be more usable when changing the appearance of text. Whether or not one product is better than another is based on the tasks users intend to perform with it.

While the design of a product may have been focused on one set of tasks, users may expand on this set in actual product use. For example, a spreadsheet application originally intended to support bookkeepers may be used by a teacher to record marks from student assignments (Carter, 2004).

The context of use may also influence whether the product is usable (Carter & Fourney, 2004b). For example, observing system designers using a system intended for use by administrative staff may say little about its usability for those administrative staff. The system designers' knowledge, background, and approach to computer systems, and thus the quality of use they attain with the system under test, are likely to be very different from those of the administrative staff, as is their knowledge of administrative tasks (Macleod, 1994).

ISO 9241-11 and ISO 13407 provide direction in how to approach usability (ISO, 1998a; International Organization for Standardization [ISO], 1999a).

2.1.1.1 Usability in ISO 9241-11

ISO 9241-11 describes usability as a composite of effectiveness, efficiency, and satisfaction with which specified users achieve specified goals in particular environments (ISO, 1998a). It recognizes that there are many possible measures of effectiveness, efficiency, and satisfaction and that all measures are subjective to some extent. Rather than using measures as absolute ratings, they can be used to compare between alternatives, such as between an existing system and a new system being developed or between two alternative new systems. The Common Industry Format for

Usability Test Reports (CIF) (American National Standards Institute [ANSI], 2001), which has been internationally standardized via ISO/IEC 25062 (International Organization for Standardization [ISO] and International Electrotechnical Commission [IEC], 2006a), provides examples of metrics related to each of these three components of usability.

Effectiveness: ISO 9241-11 defines effectiveness as, "the accuracy and completeness with which users achieve specified goals" (ISO, 1998a). Users should be enabled to use the system to accurately complete an identified set of tasks with particular types of content in various identified environments. The specific user groups, tasks, content types, and environments involved can be identified through requirements analysis. Examples of effectiveness metrics include: completion rates ("the percentage of participants who completely and correctly achieve the goal of the task"), errors (a classification of errors that prevented test participants from completing the task or that required them to "attempt portions of the task more than once"), and assists (instances where participants required assistance to be able to complete the task) (ISO & IEC, 2006a).

Efficiency: ISO 9241-11 defines efficiency as, "the resources expended in relation to the accuracy and completeness with which users achieve goals" (ISO, 1998a). Users hope that new systems will improve their efficiency in accomplishing their tasks. A system should be at least as efficient as its competition and/or any system that it intends to replace. Examples of efficiency metrics include: task time ("the mean time to complete each task") and the completion rate to mean time-on-task ratio ("the percentage of users who were successful for every unit of time") (ISO & IEC, 2006a).

Satisfaction: ISO 9241-11 defines satisfaction as, "positive attitudes to the use of the product and freedom from discomfort in using it" (ISO, 1998a). If given the choice, users will often choose the systems that they are most satisfied with. A system should be at least as satisfying as its competition and/or any system that it intends to replace. The CIF refers to a number of widely used questionnaires as a source of metrics for evaluating satisfaction (ISO & IEC, 2006a).

2.1.1.2 Principles of User-Centred Development

ISO 13407, *Human-centred Design Processes for Interactive Systems*, provides guidance on human-centred design activities throughout the life cycle of interactive computer-based systems. It is a tool for those managing design processes and provides guidance on sources of information and standards relevant to the human-centred approach. ISO 13407 describes human-centred design as a multidisciplinary activity, incorporating both human factors and ergonomics knowledge and techniques (ISO, 1999a).

ISO 13407 presents a set of four "principles of human-centered design" that characterize the commonalities from among various approaches to user-centred (or human-centred) development (or design) (ISO, 1999a). The list below summarizes the principles from ISO 13407. To improve their clarity and usability, the first principle has been split into two separate principles (Carter, 2004).

The system must be designed to meet the specific requirements of different groups of users. This includes performing some identifiable set of tasks using some identifiable types of content within some identifiable context(s) in manners that are suited to the unique characteristics of each group of users.

Users should be actively involved in the development. Meeting user requirements is more important than involving them, but it is usually very difficult to develop a user-centred system without the active involvement of some users.

Iteration is essential in obtaining and applying successive evaluations in the development of usable systems. Without evaluation there is no way to determine if a system is even slightly usable. With evaluation it is often possible to see how a system's design (and resulting usability) could be improved.

User-centred design benefits from applying a multi-disciplinary set of skills. In addition to user and software development (software engineering) skills, there are many other important skills, including (but not limited to): human-computer interaction skills, graphics design skills, cognitive engineering skills, application specialist skills, marketing skills, and management skills.

User-centred design results in an appropriate allocation of function between users and technology. Human-computer interaction, by definition, requires both the human and the computer to play a part in accomplishing some task. Usability requires that each play a part that is appropriate to their capabilities and that meets the requirements of the users.

2.1.2 Universal Access Reference Model (UARM)

Reference models of human computer interactions should "provide a generic, abstract structure which describes the flow of data between the user and the application, its conversion into information, and the auxiliary support which is needed for an interactive dialogue" (Lynch & Meads, 1986). The UARM illustrates the major functions and relations that are common to all instances of universal access, without forcing a particular design/implementation on any individual instance. This section explains the basis upon which the components and relationships relevant to the UARM were identified and is based on material that the author has developed and published (Carter & Fourney, 2004b).

The UARM was originally developed to identify areas requiring further accessibility guidance in International Standards. It identifies a range of different functions that a system must provide to support accessibility for all. Originally these UARM functions were used to analyze the guidance contained in ISO/TS 16071. This analysis identified both further guidance to add, and a possible structure that could be applied, to future versions of ISO/TS 16071 as it evolved towards becoming an international standard (i.e., ISO 9241-171).

2.1.2.1 Systems

Systems are the traditional focus of accessibility. ISO/TS 16071 concentrates on software systems, which can be used as parts of various computing devices. At this point, "system" will refer to generic systems, without specifying the type or composition of these systems. Even if the goal is for computer systems to take the responsibility for being accessible, consideration of more than just the system is necessary.

According to Stephanidis, "Universal access refers to the global requirement of coping with diversity in: (i) the characteristics of the target user population (including people with disabilities); (ii) the scope and nature of tasks; and (iii) different contexts of use and the effects of their proliferation into business and social endeavours" (Stephanidis & Savidis, 2001). Each of these aspects are considered within the UARM.

2.1.2.2 Users

Users are people who need or wish to use a system. Users will use a system because it provides some service that resolves a problem or completes a task. Any inability on the system's part to meet the user's needs may keep the user from completing their task and achieving task-related goals. For this reason, users are the main component of the model with specific concerns for accessibility. Universal access is important because no two users are exactly alike and an individual user can change over time (Carter & Fourney, 2004b).

However, as noted in Chapter 1, meeting the needs of users and universal access is often seen as a burden rather than a design goal. Avoiding an accessibility view focusing blame on users or systems, allows seeing a third party involved — the interaction between the user and the system.

2.1.2.3 Interaction

The term "interaction" describes the process of communication (or negotiation) that occurs between a user and a system. Interaction is what ties systems to users. Users interact with a system to accomplish a task or set of tasks. Interactions can be considered the tangible components of a task. Tasks may involve a series of individual interactions. Interactions occur in two directions: from the user to the system and from the system to the user. Interactions occur by sending messages across channels.

Interactions are not necessarily serial in nature. Multiple interactions, especially by systems to users, may occur at one time. Multiple interactions may make use of the same or different forms of communication and may reinforce or even duplicate one another. Redundant communication may provide greater potential for universal access but may also provide an unnecessary load on the user.

Any failure of one or more interactions to meet the needs of the system, task, and user can result in accessibility problems. Since determining and removing the cause of such a failure can improve accessibility, interactions should be the focus of accessibility.

2.1.2.4 Handicaps

Handicaps are anything that may interfere with the accessibility of interactions. As noted in Chapter 1, a handicap is, "the loss or limitation of opportunities to take part in the normal life of the community on an equal level with others due [to] physical or social barriers" (DPI, 1993).

A handicap may have one or many sources among the system, user, interaction, and/or environment. Understanding the handicap to the interaction is more important than attributing the blame. Multiple handicaps may be present during a set of interactions. For example: consider an individual attending a lecture. There are many potential sources of handicaps to the interaction. The lecture may be presented completely aurally and the individual may have a hearing disability (which could be considered a user-related handicap to the interaction). The speaker may not speak clearly (which could be considered a system-related handicap to the interaction). The individual might not understand one of the points being made and may not be allowed to ask immediately for the necessary clarification to continue to understand the lecture (which could be considered an environment-related handicap to the interaction). If the individual received an urgent telephone call in the middle of the lecture or if some loud noise made hearing difficult, some of the lecture might be missed (which could be considered another environment-related handicap to the interaction).

Figure 2.1 shows a first attempt at a reference model. It uses the metaphor of a valve to illustrate various levels of interference from a handicap. A fully open valve would represent no interference. A fully-closed valve would represent full interference. Any other setting of the valve would be a partial interference. Just as handicaps can act to restrict interactions, there are other factors that can assist interactions. These factors are referred to as contexts.

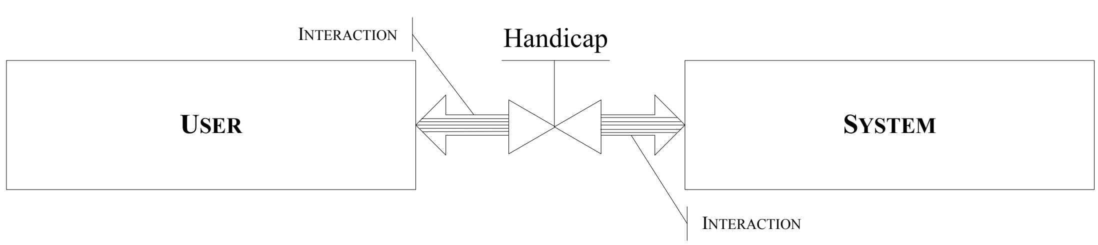


Figure 2.1: A first attempt at a reference model (adapted from Carter & Fourney, 2004b)

2.1.2.5 Contexts

Users and systems have their own contexts that they use to interpret and fill in gaps in the interactions that they receive (Maskery & Meads, 1992). Context can help to reduce the effect of handicaps that only partially inhibit communication. However, if the interaction is fully handicapped, as indicated in the model by a fully-closed valve, there is no communication for the context to interpret or fill in. In this case, some direct change must be made in the source of communication to at least partially remove or avoid the handicap to the interaction first, before trying to use context.

Contexts are, and should be, shared between users and systems. For any interaction to be successful, it requires the user and the system to use some shared context (i.e., symbols/language and/or application knowledge) to make sense of the interaction. For example, both the system and the user may use their context/knowledge of the English language to facilitate communicating textual messages. As illustrated in Figure 2.2, this shared context provides a long-term link between the user and the system.

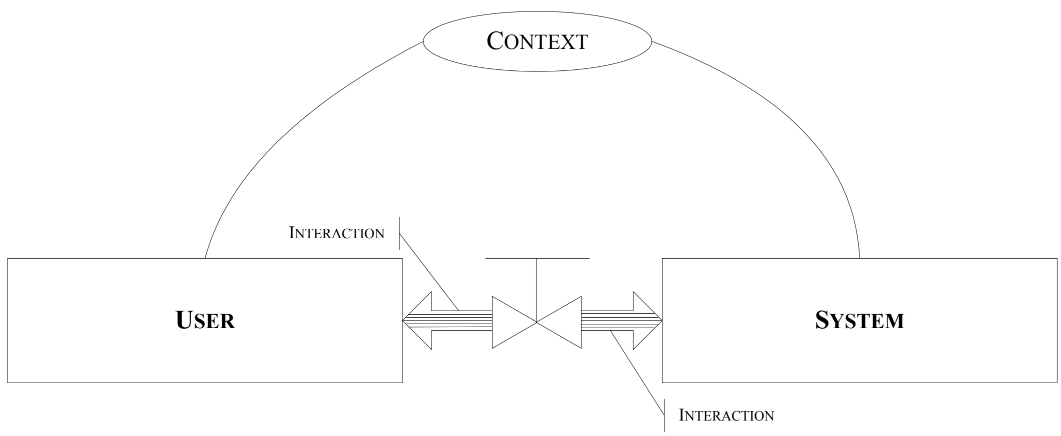


Figure 2.2: Adding context to a reference model (adapted from Carter & Fourney, 2004b)

If the interactions provided by the system are appropriate for the user of a task in a given environment, there will be no handicaps.

The necessary content of interactions may be reduced by implicit or explicit reference to a context that is expected to be shared between the user and the system. When this expectation is not met, a contextual gap occurs. The use of shared context may also reduce the effect of handicaps on current interactions.

All interactions are interpreted relevant to the context of the receiver. Where contextual gaps occur, further interactions making use of a shared context is necessary to clarify the original interaction.

Contexts may hinder (as well as help) interactions if they are not appropriate to the interaction. Where context is used inappropriately (such as with colloquialisms that are not shared and are thus taken literally) it may lead to misassumptions. Where context is required but missing, its absence will handicap the interactions. The specific environment in which an interaction occurs can focus attention on specific contexts.

2.1.2.6 Environments

Environments provide additional contexts that focus the user or system on particular portions of their own contexts. Users, interactions and systems may not share the same environment and may each exist within multiple competing environments.

An environment may be physical or socio-cultural. Physical environments include built spaces, such as homes, offices, and mechanical plants, as well as the effects of a user's own physiological state. Changes to a user's physiological state may be short-term, long-term, ongoing, or permanent. Examples include hunger, illness, progressive loss of sight, or permanent injury. Social and cultural environments include wide-spread attitudes towards the system being used.

The term "handicap" emphasizes the, "shortcomings in the environment" (United Nations Commission for Social Development on Disability, 1993). An environment may help or handicap the interaction. It may help the interaction by suggesting the context to use when interpreting the interaction. It may handicap the interaction by introducing distractions and/or incorrect and/or inappropriate suggestions for context. In addition to focusing attention on existing context, environments are a source of additional context. Contextual knowledge of the environment of the user may help the system respond to the user more appropriately.

The relationship between the environment and both context and handicaps completes a high-level view of the UARM, as illustrated in Figure 2.3.

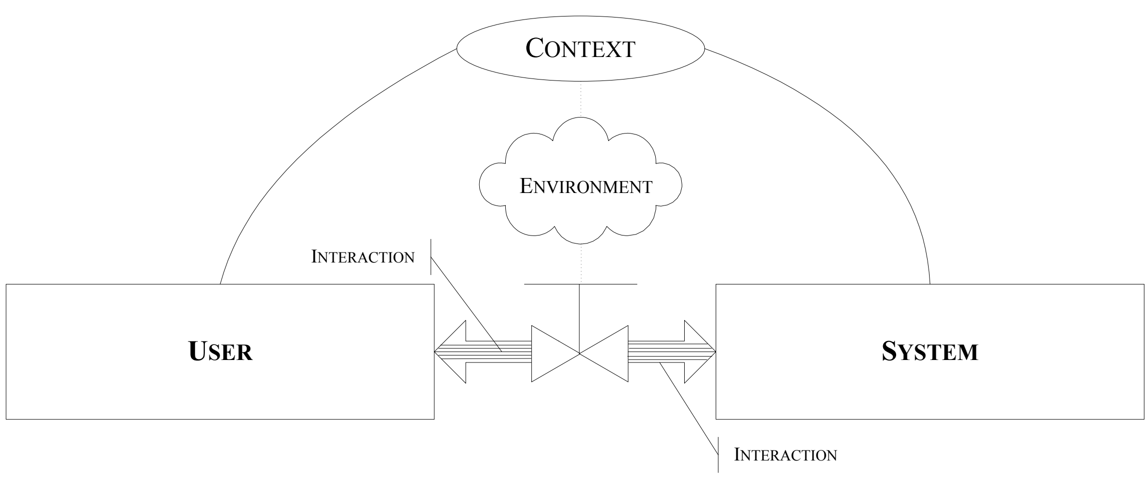


Figure 2.3: The "Universal Access Reference Model" (adapted from Carter & Fourney, 2004b)

2.2 Further Considerations on UARM Components

This section builds on the previous introduction to the UARM by describing the components of the model in greater detail. The model contains five components: users, systems, interactions, environment, and context.

2.2.1 Users

A "User" is a person who interacts with a product, service, or environment (ISO & IEC, 2001b). Differences between users and differences experienced by a single user over time provide different accessibility needs. Different users have different capabilities and/or preferences, as well as different task-related needs when using the same system. Short-term, progressive or permanent changes to one's abilities, skills and/or preferences means that each individual user may have different accessibility needs at different times.

User interactions involve the basic functions identified by Communication Theory (Shannon, 1948). As such, users:

- encode/decode messages using context that they hopefully share with the entities with which they are interacting,
- transmit/receive messages using various media.

Figure 2.4 illustrates the major user functions involved in encoding/decoding and transmitting/receiving messages. The actual transmission/reception of messages involves a combination of physical functions in the user's interface with the real world. However, each physical function requires a corresponding skill/ability contained within the user's context. The user's context (a combination of abilities, skills and preferences) serves as the decoder/encoder. The user's mind in this model is the source and destination of interactions.

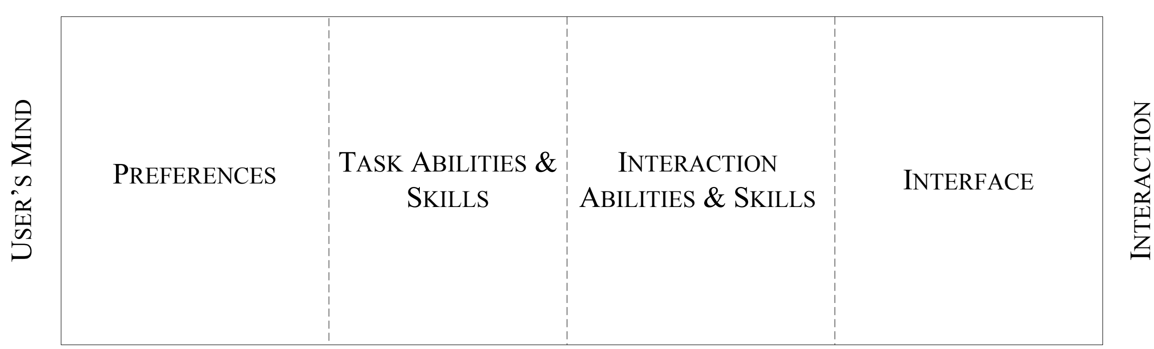


Figure 2.4: A model of a user (adapted from Carter & Fourney, 2004b)

2.2.1.1 The User's Interface

The function of the user's interface, in this model, is to create, select and manage the combination of channels (through the user's physical senses) that are used to interact with the outside world. The interface creates output channels and selects input channels based on the user's abilities and skills. Each channel requires certain skills to be used successfully.

The interface provides the synergy needed across various interaction channels to allow the simultaneous transmission and reception of multiple messages. It also allows the user to focus on particular messages or channels. The communication possibilities and the needs of a user at a given time may be greater than the user's capabilities. For example, users are seldom capable of receiving information via multiple audio channels simultaneously. A user only capable of receiving information via audio cannot receive as much information at the same time as a user capable of receiving both aural and visual information.

The interface's management function optimizes the set of channels in use to help the user avoid information overload. This includes filtering or ignoring various competing interaction channels, as well as noise within channels.

2.2.1.2 The User's Profile

Users are complex beings with many interacting skills and preferences. The need of interfacing with a particular user is based on a comprehensive set of abilities.

Bloom's Taxonomy (Bloom et al., 1956) identifies a widely accepted set of abilities that is useful in evaluating user behaviours. The taxonomy classifies abilities along three domains of activities {*cognitive*, *affective*, *psychomotor*} and allows them to be measured along a continuum from plain and simple to rather complex. The UARM recognizes these domains within a user's skills as they directly map to the filters a user applies to enable/disable the channels a user chooses to (or must) use to interact with a system. A user's preferences maps to the affective domain, task abilities and skills to the cognitive domain, and interaction abilities and skills to the psychomotor domain.

Bloom's taxonomy also maps to the usability objectives of ISO 9241-11. Satisfaction, an emotional response, maps to the affective domain. Efficiency, a metric of skill, maps to the psychomotor domain. Effectiveness, a metric of accuracy and completeness, maps to the cognitive domain. In this sense, Bloom's taxonomy of affective, cognitive, and psychomotor capabilities provides a justification of the metric provided by ISO 9241-11.

Several authors (Edwards, 1995; Jacko & Vitense, 2001; Carter & Fourney, 2004b) also discuss a large number of cognitive, perceptual, physical and psychomotor abilities that make up many of the interaction abilities and skills suggested in the UARM.

The UARM identifies three sets of user characteristics that can be combined to profile the user. A user's profile involves three types of abilities that enable or disable the channels a user chooses to (or must) use to interact with a system.

Interaction abilities and skills allow a user to interact using a particular communication channel. Such skills include a large number of cognitive, perceptual, physical, and psychomotor abilities such as physical extension or dimensions (e.g., hand size, height), education, and literacy. One's interaction abilities and skills fluctuate as they may be affected by the onset of, and recovery from short-term or temporary disabilities and illnesses. Within the UARM, the set of interaction abilities and skills are recognized to change over time and in different circumstances. While permanent or temporary physical disabilities may impact certain user interaction abilities, external factors, such as environmental noise, may also handicap the interaction. Within the model presented here, the set of interaction abilities and skills are recognized to change over time and in different circumstances. Thus, what is important to consider is that interactions must use channels that the user currently has the interaction abilities and skills to use.

Task abilities and skills allow a user to make sense of the content of interactions. Tasks provide purposes and understanding for interactions. Tasks are accomplished using particular cognitive capabilities that have been specially developed. Tasks and/or their content may require (or prefer) the use of certain interaction abilities and skills related to certain media types or channels of communication (International Organization for Standardization [ISO], 2002a).

A user's personal preferences can affect the choice of channel used wherever various channels are available for some interaction. A person's state of mind has an impact on attention and performance. Attentiveness can affect efficiency and performance on workload demands (Taubes, 2000). A person's preferences and habits are learned behaviour and/or reflective of the personality and mental model of the user. A user's preferences and habits will change how an activity is completed and achieved by different users even if the task, environment and characteristics of the users and situation are the same (Inclusion of Disabled and Elderly People in Telematics, 2000). Cultural differences may influence a user's preferences. User preferences may also change depending on the user's current emotional state.

The user's unique set of abilities may hinder or benefit the interaction between the user and the system. From the user's perspective, an accessible interaction is only possible if there are sufficient channels available that are supported by the user's interface and context. When compared to the set of channels the system makes available to the user, those abilities that the system and user both share are available for successful interactions. Therefore, shared abilities will open channels, while unshared abilities will close channels.

2.2.1.3 Life Changes

During the span of one's life, one may experience any or all of the following six changes (which are not necessarily negative):

Birth: Birth is a point in a person's life where one would not have an experience of change. In particular, a person would not have an experience of "norm." It is for this reason alone that many persons born with a disability tend to describe themselves this way only because others do, they have not actually *experienced* whatever ability had been "lost."

Accident/Illness: An accident/illness is a sudden, possibly temporary, change in one's physical status/health. For the person with the accident-related injury or an illness, it is easier to recognize that a change in status has occurred; that is, they differ from both a "social norm" and their "personal norm" (i.e., body schema Reed & Farah, 1995) Since the person is aware of the change, this requires the person to find assistance with the systems they use and develop coping skills for the changes they have experienced.

Ageing: Ageing (infant to child, child to youth, youth to young adult, young adult to middle age, middle age to elderly) is a gradual ongoing change in a person's life. Unlike the sudden change brought on by accident, ageing requires recognition that a change has occurred and acceptance of the change. It is always possible that a person experiencing this gradual change is ignorant of the change; such a person may not realize that a system requires adaptation. In addition, coping mechanisms may be slow to develop.

Recovery: A person's recovery from an illness or an accident-related injury will also change their coping mechanisms and abilities, though not necessarily to their previous status.

Assistance: A part of a person's coping strategy may be the acquisition of some assistance. This may be anything from hearing aids to personal care assistance. Almost always, it is a partial remediation of the person's needs; that is, for a given individual, there will be things that are still not done. With the acceptance of assistance, there may be negative consequences such as the perception by others that, with this assistance, there is no disability. For example, others may ignore the need to face a hard of hearing person when communicating simply because it is assumed that a person wearing hearing aids no longer has a hearing problem.

Learning: Learning is a lifelong activity with broad impact. Over the span of a person's life, change due to formal education and personal discovery also occurs. Such change is not necessarily gradual as learning can occur both over long periods as well as short "eureka" moments.

The presence/absence of an ability should be seen as a continuous function rather than as the basis for dividing users into distinct groups of "haves" and "have-nots" (Vanderheiden, 1990). The distribution of skill in a specific ability can be described like a curve which includes a small number of those who have an exceptionally high skill, a larger number with mid-range skill, and a long tail of those with little to no skill.

> Thus, individuals do not fall at the lower or upper end of the distribution overall, but generally fall into different positions depending on the particular ability being measured. (Vanderheiden, 1990)

2.2.2 Systems

Traditional human-computer interaction models of software systems may be divided into three parts: a front-end interface, the application (processing) logic, and a back-end database. The model, illustrated in Figure 2.5, expands on this basic structure by including interaction components, which are used to provide interaction styles and media necessary for an accessible interface.

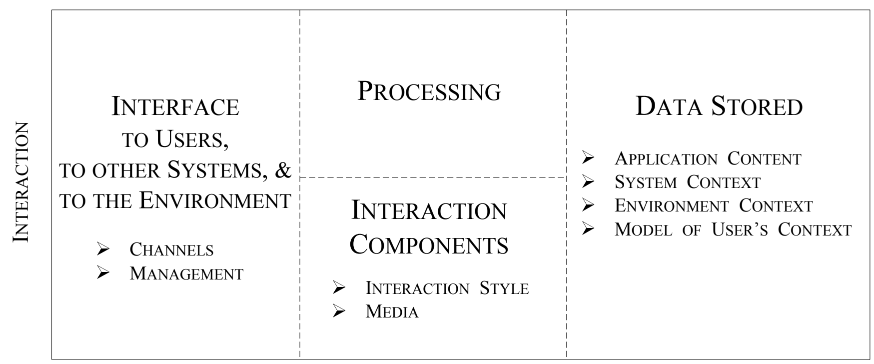


Figure 2.5: A model of a system (adapted from Carter & Fourney, 2004b)

Note the similarity between Figure 2.5 and Figure 2.4. In terms of Communication Theory (Shannon & Weaver, 1949), the Interface would be the receiver/transmitter, the Interaction Components together with part of the Processing functionality the decoder/encoder, and the Data Stored together with part of the Processing functionality is the destination/source of all messages.

The remainder of this section discusses each of this model's features.

2.2.2.1 The System's Interface

A system's interface is composed of a number of channels that provide the input and output functions of the system which interact with the user either directly or via one or more ATs. Individual channels correspond to particular hardware devices that are part of the computer system.

Systems interface with users, for example, through the screen using a graphical user interface. Systems interface with other systems, such as ATs, for example, through connections to these systems and/or their processing component. Systems interface to the environment, for example, through sensors such as temperature controls.

Successful interaction requires that the user's interface be capable of interacting with the system interface and/or any ATs connected to this interface. The system interface should be aware of, and support, any ATs that are being used to increase accessibility.

The system's interface should manage its combination of channels to achieve a suitable synergy that fulfills the needs of both the user and the system's processing. This management function should provide sufficient interactions to meet these needs without providing information/interaction overload. This management function should also take into account any ATs that are being used, so as to maintain synergy as much as possible.

2.2.2.2 Processing

The system's processing functionality executes the application's logic for the system. It performs this function by interacting with the data store, making use of interaction components and interacting with the user through the system's interface.

The application logic specifies how the system will assist the user in performing a variety of (application) related tasks. As already noted, these tasks and/or their content may require (or better suit) the use of certain types of channels. However, to maintain accessibility, the channels presented to the user must correspond to the user's interaction components.

An additional accessibility concern is that the system's tasks should be compatible with the user's task abilities and skills. If the user does not have the required task abilities and skills, then the user must be able to acquire these from the system (through help and/or tutorial functions) before making use of system processing functions involving these tasks.

2.2.2.3 The System's Interaction Components

The system's interaction components provide the basic interaction styles and media that can be used by the system's processing functionality to produce the system's interface. Each interaction style and type of media may be used any appropriate number of times within the resulting interface.

By making use of standard interaction styles and media, it is easier to support accessibility needs either directly or through the use of ATs. Each interaction style and media type has its own accessibility issues that need to be taken into account. The major styles of interaction, except for natural language and gesture, have been standardized within the ISO 9241 series; however, none of these standards contain specific accessibility-related guidance. Web-based interaction styles are standardized within standards set by the World Wide Web Consortium (W3C), along with accessibility-related guidance set out in the Web Content Accessibility Guidelines (Chisholm, Vanderheiden, & Jacobs, 1999).

Interaction styles are rendered through media to produce the channels that the system interface provides to the user.

2.2.2.4 System Stored Data

A system may have up to four core pieces of data. Two, Application Content (including any data and task content) and System Context (i.e., its own context) are required by the system. The other two, Environment Context and a Representation of a User's Context, while somewhat more "optional," have a direct impact on improving the interaction shared by the user and the system.

Application Content

Application content is the system's knowledge of the application domain and includes the data being used and the task being performed. The data being used by the application is a user data storage component (e.g., in a word-processing system, the current document). Information about the task is knowledge of what the application is doing (e.g., word-processing).

Being able to provide more than one means of system interaction may require the application content to be separated from the tool itself. An example of this is a browser which presents the information (i.e., a web page) retrieved from a remote source but the information itself has no effect on the browser. The web page content can provide its own full interface that is separate from the tool thus providing more than one means of system interaction.

Knowledge of the application content maps directly into a system's processing functionality. Application content is used by the system's processing functionality as the system processes information and cooperates with the user to complete the task.

System Context

A system maintains information about its own context. A system's context knowledge includes information about available interaction styles and media, and current system state. Knowledge of the system's context maps directly into the system's interaction components. System context is used by the interaction components to define available interaction styles for the interface.

Environment Context

A system may maintain information about specialized contexts such as its own environment context. The environment context is one of the two types of data that the system might not have. Knowledge of the system's environment context maps directly into the system's interaction components.

User Context Model

A system may contain a representation of the user's context. The user context model is the other of the two types of data that the system might not have. A system's user context model contains an approximation of the user's interaction, task, and preferences abilities and skills. This information assists the system in cooperating with the user to complete the task. This representation may be based on current or previous interactions with the user, and/or derived from an analysis of the channels the user has chosen.

2.2.3 Interactions

Accessibility depends on the recipient's ability to create, receive, and interpret interactions. Interaction involves a number of one-way messages between users and systems and between systems and users. The theoretical approach to the definition of an "Interaction" can be found in Communication Theory (Shannon, 1948).

From the perspective of Communication Theory, the interaction between the user and the system can be described as one-way and transmission-oriented (see Figure 2.6). In this sense, any difficulties (i.e., "noise" in the language of Communication Theory) in the communication channel would inhibit the effective transmission of the system's content (i.e., the application or service the system provides). The environment is seen as one source of such noise (Shannon, 1948).

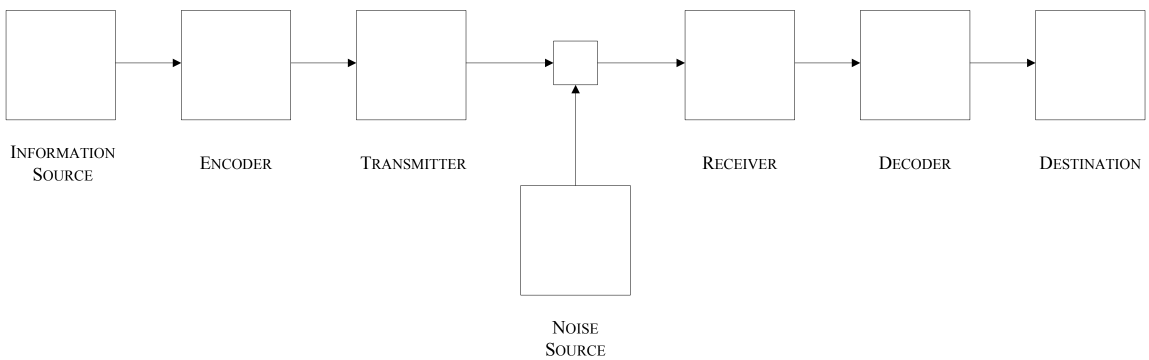


Figure 2.6: "Communication Theory" view of interaction (adapted from Shannon & Weaver, 1949)

Shannon's Communication Theory presents a number of important functions that must take place between two entities attempting to communicate with one another (Shannon, 1948). A message travelling between its source and destination must be:

- encoded (which should be done using a shared context),
- transmitted via a specific medium,
- over a communication channel connecting the sender and the recipient; that is, subject to noise (interference),
- received by its intended recipient (if the recipient is capable of using that medium), and
- decoded (using the recipient's version of shared context).

If the context of the recipient is known, the choice of media can be limited to those that the recipient can access successfully. Accessibility may be increased by using multiple messages transmitted via different media over different channels, so the likelihood increases that one or more messages will be received and interpreted.

Each channel may have its own set of handicaps, as illustrated in Figure 2.7. Handicaps to the interaction may occur where the recipient of a message is unable to make use of a message, and where there is no alternative or redundant message available. In addition to other contextual considerations, this may be due to the medium or the channel of the message. The recipient's capacity to use a medium can be considered a recipient skill and thus a part of the recipient's context. Environmental noise may interfere with interactions using one or more media, and can create even more handicaps to the interaction. For example, a user's noisy environment may make information given through an auditory channel useless or make receiving information given through an auditory channel very difficult.

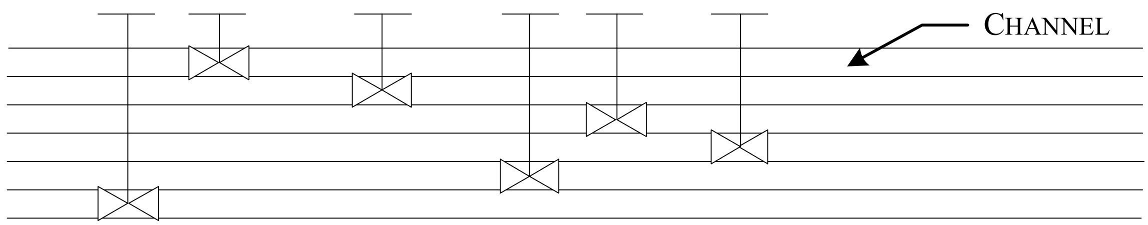


Figure 2.7: Media channels and handicaps

It is important to remember to focus the consideration of handicaps as they limit interactions, rather than individual messages. Interaction is a two-way process involving a number of one-way messages. These one-way messages may or may not be asynchronous. Users interact with a system to use the system; systems interact with a user to respond to user requests.

From the user's perspective, an accessible interaction is only possible if, after filtering the available channels against the user's real skills, there remains some set of channels that allows full interaction. If there are not enough channels available for full interaction, partial interaction may still be possible. However, if there are not enough channels remaining for a user to access a system, the system may be completely inaccessible to the user. Since accessibility is dependent on the message recipient's ability to receive, interpret, and create interactions, accessibility may be increased by transmitting multiple messages via different media over different channels. Hopefully, one or more messages will be received and interpreted. Each medium makes use of its own communication channel. Messages may be combined with one another and/or transmitted simultaneously. One of the roles of these messages is to provide a means to share context.

Handicaps affect interactions, rather than individual messages. If the interaction succeeds, in spite of difficulties with individual messages, then accessibility will be maintained. However, for the sake of efficiency, it is preferable that only successful messages be involved in the interaction. This can be accomplished if the sender of the message is aware of the capabilities of the intended recipient of the message. In the UARM, this involves being aware of shared context.

The single valve, in Figure 2.3 on page 19, is a simplification of what is really happening. Each channel is uniquely influenced and thus can have its own handicap to the interaction. As Figure 2.7 shows, it is more correct to depict each channel as having its own valve.

A channel is a function of media, style, and content, and results in usability.

Media

A medium is a way of rendering information to a user. Each medium can be classified in terms of the modality it supports. A modality is the type of sensation the medium uses (e.g., visual, auditory, tactile, olfactory). Each modality only supports content in specific media. This will be discussed in Section 2.4.1.

Style

Media render interactions that are based on particular dialogue styles. The ISO 9241 series contains guidance on a variety of dialogue techniques that form the basis for different styles of interaction. The standard interaction styles in application software and operating systems are: menu (International Organization for Standardization [ISO], 1997a), command language (International Organization for Standardization [ISO], 1997b), direct manipulation (International Organization for Standardization [ISO], 1999b), form fill-in (International Organization for Standardization [ISO], 1998b), natural language, and gesture (Carter & Fourney, 2004b). ISO 14915-2 provides further information on controls and links that may be implemented through different styles (International Organization for Standardization [ISO], 2003a).

Content

The actual content of an application can be composed of several different media-neutral types of information including: causal, conceptual, continuous action, descriptive, discrete action, event, physical, procedural, relationship, spatial, state, and value (ISO, 2002a).

Usability

Usability is the result of combining media, style, and content in a channel such that it is effective, efficient, and satisfying for the user (ISO, 1998a). The effectiveness and efficiency of a system for a user is dependent on the choice of media, style, and content used to communicate across a channel. Handicaps to interactions directly influence a system's usability. The presence of a handicap and the degree it interferes with the channel will impact system effectiveness to the point that one's objective cannot be achieved.

2.2.4 Environment

Users interact with systems in an environment. Environments may affect users, systems, and interactions by helping or handicapping the interaction. It helps the interaction by suggesting the context of use when interpreting the interaction. It handicaps the interaction by introducing distractions and/or incorrect and/or inappropriate contexts of use.

Environments may be physical and/or socio-cultural. Physical environments include built spaces such as homes, offices, and mechanical plants as well as an individual's physiological state. Social and cultural environments include wide-spread attitudes towards the system being used.

Users, systems, and interactions may share the same environment but each be affected in different ways. For example, an environment high in electromagnetic noise may not affect the user, but it could affect the system. Similarly, an environment filled with noxious olfactory noise (i.e., a really bad odour) may dramatically affect the user but have no effect on the system.

Users, systems, and interactions may not share the same environment; each may have a different separate environment affecting them in different ways. During the period of an interaction, the environment of the user may not be the same as that of the interaction. Simultaneously, the environment of the interaction may not be the same as that of the system. Thus, knowledge of the environment of the user may help the system respond to the user more appropriately.

There is a temporal dimension to an environment. An environment might change during the period of the interaction. Such change could be instantaneous, rapid, or over a period of time. For example, a room might become ten degrees cooler over an hour. Similarly, a water pipe might suddenly burst.

2.2.4.1 User Environment

Although no user or system operates in a vacuum, a single perceived environment may not be shared by both system and user.

For a user, the various factors of their environment may assist or hinder their performance with a system. For example, the user's comfort in the environment (e.g., too hot, too cold), ability to get around the environment (e.g., an accessibly built environment, cleanliness/neatness of space), and ability to focus on the one system (e.g., divided attention to other tasks, presence of noise[1]) each contribute to the user's performance.

A user's physiological state is part of their physical environment. One's physiological state

[1] "Noise" in this context may be visual, olfactory, auditory and/or tactile.

provides a constant flux that frequently impacts one's well-being. Such changes to one's physiological state may be short-term, long-term, ongoing, or permanent. Examples include being too hot, hungry, sick, or permanently injured.

A user's socio-cultural milieu is also part of their environment. For example, while using a kiosk system, the presence of a potential user in the line behind the current user could distract the current user from their task. An even longer queue might provide further distraction.

2.2.4.2 System Environment

Knowledge of the environment of the user may help the system respond to the user more appropriately. Knowledge of its own environment may also help the system when responding to the user.

A system may maintain information about its own environment. Such information may be "perceived" by the system through various sensors (Culler, Estrin, & Srivastava, 2004), system alarms (Tohma, 2004), or hardcoded information (e.g., "system preferences"). Sensors, such as thermostats, if present, can provide information about the system's external space. System alarms, if present, can provide information about internal hardware or software status. Hardcoded information can be provided by a programmer, maintainer, or user. Hardcoded information may only be entered once, updated regularly, or updated irregularly. However, unlike sensors or alarms, hardcoded information may not correctly reflect the average physical environment of the system; only a snapshot in time.

A system may store a model of any information it has about the user's environment. This information assists the system in co-operating with the user to complete the task. Information about the user's environment may be based on current or previous interactions with the user, and/or derived from an analysis of the channels the user has chosen to use to interact with the system.

2.2.5 Context

Context is the "glue" that ties interactions, systems, and users together. Each party in an interaction has their own context.

The theoretical basis for context in the UARM is provided by Weaver's notion of context in Communication Theory (1949). The problem that context identifies is that each party in an interaction can have their own perspective, which may help or hinder the success of the interaction. This includes the system which must share its own expectations with users if it is to successfully mediate any interaction.

For any channel of transmission to be successful, it requires the sender and receiver to have a shared context — something in common — for the encoding/decoding of messages (Weaver, 1949).

For context to be used to "make sense" of any message, there must be some common knowledge related to the message (e.g., English) and some common skills that help make use of the channel (e.g., literacy). In the case of a computing system, both the user and the system:

- must share knowledge of the application domain,
- may have domain-related skills (Carter & Fourney, 2004b), and
- must share knowledge of the symbols used in the communication (e.g., communicating in Arabic requires both knowledge of Arabic and literacy in oral/written Arabic) (Weaver, 1949).

From Weaver's approach, the shared context of the interaction requires:

- including the interaction skills for non-computer environments into the user's context (i.e., remembering that the user is not just interacting with the system, but also their own physical environment); and
- recognizing that a user can be involved in multiple interfaces simultaneously such that the user's full attention cannot be assumed.

Therefore, to communicate, all parties must be able to anticipate something within a shared context. When context is missing, more information must be provided. Anything that is not shared has to be communicated or explained in a way that is shared.

From this point of view, the user and system exist in many overlapping contexts (Maskery & Meads, 1992). Each of these contexts can be logically AND'ed or OR'ed such that the interaction between the user and the system can be described in the form of a Venn diagram (see Figure 2.8). Each context represents information about the task at hand, the environment, and so on. Any unshared portion of each party's context is a potential area for misunderstanding and miscommunication, which may (or may not) lead to a handicap.

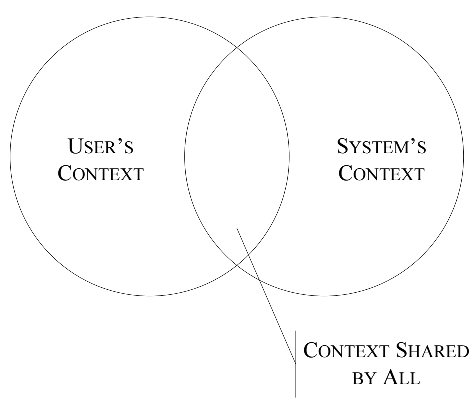


Figure 2.8: Shared context and accessibility

Context involves a combination of abilities, knowledge and assumptions. A user's context is composed of the set of abilities, knowledge and assumptions that are built up from a history of previous interactions with the same and with similar systems. A system's context is the abilities, knowledge, and assumptions it has of the task/problem. However, a system's context is based on what the developer provides it. Unless it is intelligent and capable of adapting its skills during use, a system uses this limited hard-wired context.

Three pieces of context, A, B and C are illustrated within Figure 2.9. Context "A" can increase accessibility, because it provides a shared context to help the interaction to accomplish some task. If the context necessary to interpret an interaction is not shared then the interaction is handicapped. Context "B" represents a user context that the system does not share, and which, if used by the user, may decrease accessibility, and thus handicap the interaction. Context "C" represents system context that the user does not share, and which, if used by the system, may handicap the interaction.

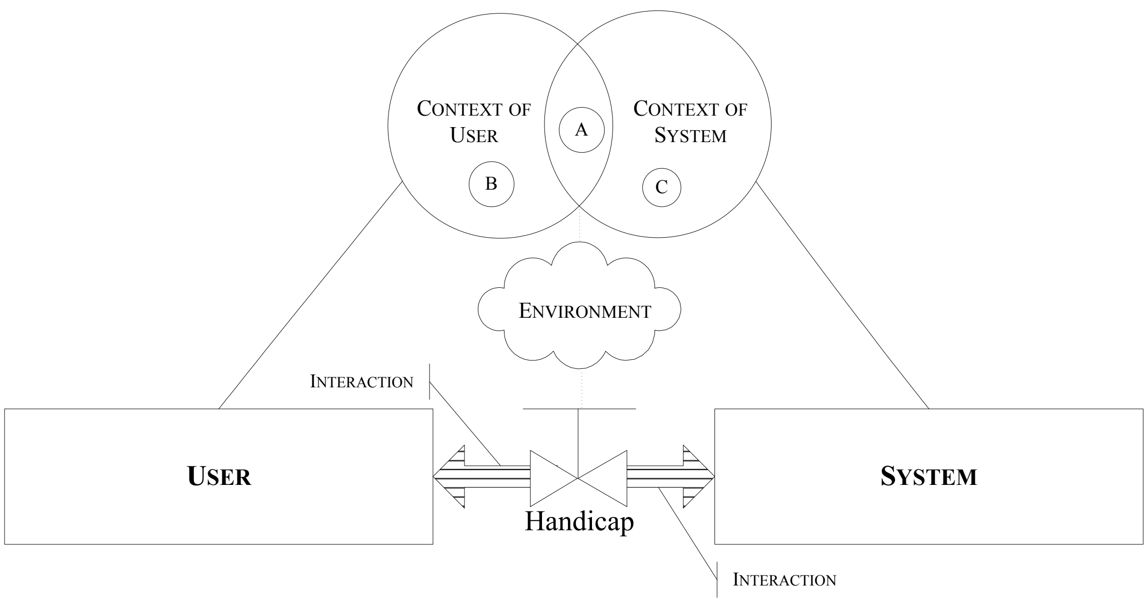


Figure 2.9: Not all context is shared (adapted from Carter & Fourney, 2004b)

Therefore, the existence of a shared context is not sufficient to ensure accessibility. Rather, it is important to ensure that a context used in interactions is shared. This requires some recognition by both the user and the system of the context that the other is using. By recognizing the shared context that exists, a non-shared context can be avoided and accessibility increased. It is important to recognize that interactions can expand the amount of shared context over time. This does not preclude users or systems from having their own individual non-shared contexts that they use for their own processing. It just stipulates that they should only use shared context for creating and interpreting interactions.

In this sense, if there is a handicap to the interaction present, the parties involved must find a shared context or create a new shared context to get the information through.

2.3 Multi-System Models of Interaction

This section introduces multi-system models of interaction. Up to this point, the chapter has assumed one system interacting with one user.

Multi-system models recognize that systems network with other systems. For example, a system can be made up of countless resources (e.g., computing and storage units) and communication facilities, which may be allocated over networks (Tohma, 2004).

Multi-system models illustrate additional systems, such as ATs, interacting with a user. Often users interact with a combination of components, each of which can act like a system on its own. Applications of multi-system models include personal area networks (PAN), a computer network close to one person (Zimmerman, 1996), and other mobile computing systems.

The UARM can be expanded into a multi-system model illustrating the presence of other systems such as ATs. One opportunity for further exploring the roles played by ATs is available within the UARM's approach of describing handicaps as a valve.

2.3.1 UARM and Assistive Technologies

An AT is an extension, program, device, and/or utility added to a system to make it more accessible to users with or without disabilities (Microsoft Corporation, 2000). This definition focuses on adaptations to products. A more comprehensive definition is provided by ISO 9241-171:

> **assistive technologies**
> hardware or software that is added to or incorporated within a system that increases accessibility for an individual (ISO, 2006f)

An AT is a system situated between a user and the actual system being used. Its role is that of a "middle man" providing a façade of the system to the user that is, hopefully, more accessible to the user. An AT acts on a channel, or a set of channels, by transforming interactions. These transformations are intended to result in accessibility improvements. As shown in Figure 2.10, an AT can be illustrated as having multiple channel interfaces with transformation functionality between their input and output channels.

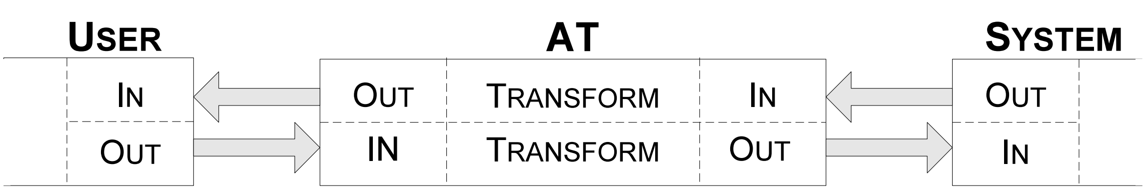


Figure 2.10: Accessibility interfacing and processing

ATs reduce handicaps to interactions. The assistance provided by an AT does not necessarily imply the *removal* of a handicap, only potential for reduction. An AT may be used because there is some component of the interaction that is “handicapping” the user. However, a handicap to the interaction does not need to be present for a user to choose to use an AT. Similarly, a consumer of an AT may not have a disability.

In the computing world, an AT can be realized through: alternative input devices (e.g., trackball, left-handed mouse, sip/puff systems), alternative output devices (e.g., voice, Braille display), and accessible software (e.g., screen magnification software). “Universal Design” (i.e., barrier-free design) can make an AT unnecessary or facilitate the inclusion of an AT. Since the interaction is what is being handicapped, an accessible computing experience is realized by a reduction of this handicap.

Although the UARM was originally developed to identify areas requiring further accessibility guidance in international standards, it is useful in outlining a user model applicable to an accessible computing experience. It can be evolved to explicitly show the role of an AT within the user/system interaction.

As illustrated in Figure 2.11, ATs function to open the valve between systems and users. For greater legibility, the effect of the environment is not displayed in Figure 2.11 as it was in Figure 2.3, on page 19. An AT serves as a proxy within the interaction between the user and the system by providing contexts compatible to each of the user and the system to perform the translation of specific media types/channels in each direction. For this reason, an AT is not illustrated as a box, but as a modified valve. The multiple contexts used and manipulated by ATs are used selectively to better match the individual contexts of users and of systems. It is hoped the AT provides a user an experience of an alternate/modified system that is similar to that of users who use no ATs and do not experience a handicapped interaction[2].

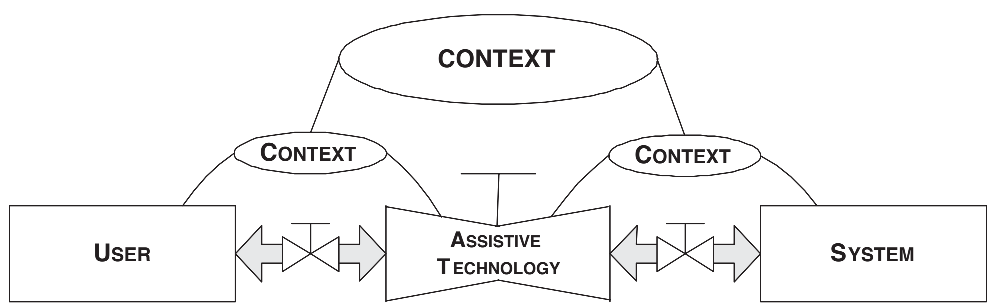


Figure 2.11: Assistive technology in the UARM

[2]The user may realize that an AT is connected to their system. However, it is hoped that the connection is seamless such that the user’s experience of the system is similar to that of users who use no ATs and do not experience a handicapped interaction.

For example, suppose a system provides a command line interface on a visual display. The system's context about users may not include users who cannot use a visual display. Further, the system may not have the ability to display the command line any other way. Adding an AT to the system modifies the system to have both context about users who can and cannot use a visual display as well as the ability to display the command line in other ways. As a consequence, adding the AT better matches the individual contexts of the user and the system.

The effect of the environment does not disappear. With the inclusion of an AT, or any other additional system, into the model, the environment or environments involved now affect more than one communication. The effect of the environment is reintroduced in Figure 2.12. When the environment acts on channels, its influence is most noticeable when it handicaps (acts negatively on) interactions.

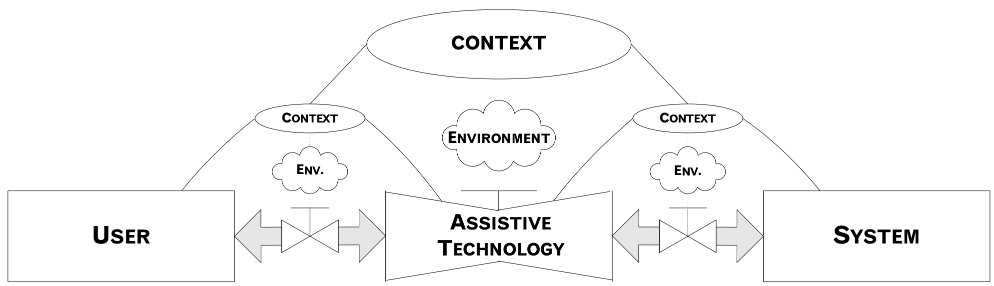


Figure 2.12: Environment in a multi-system model

As Figure 2.12 shows, there are several ways the environment is involved in the multi-system model. An AT may be added to an interaction because of a noticeable handicap in the user-system interaction due to the environment. An AT's transformed output can be noticeably handicapped by the environment, consequently nullifying the potential benefit of the AT. In addition, the AT itself, since it is also a system, can be directly affected by the environment.

ATs can be analyzed in terms of their role as special cases of systems. An AT may combine the processing, data, and interaction components parts, as illustrated in Figure 2.13.

The data part is only present when the AT is "programmable". Unlike software-based ATs, many hardware-based ATs do not use programmable data to control their operations. The stored data is used by the processing part to select appropriate interaction components.

The primary use of the processing part of the AT is to take the content received by one interface, transform the content, and then send the content to the opposite interface. This transformation is necessary to match the user's context with the system's capabilities and context.

The User-AT Interface is similar to the interface of a traditional system in that it involves a set of available channels from which a user chooses. The System-AT Interface is also like that of a traditional user in that it accesses a set of available channels provided by the system. Software-based ATs work the same way; however, the interfaces involved may be logical rather than physical.

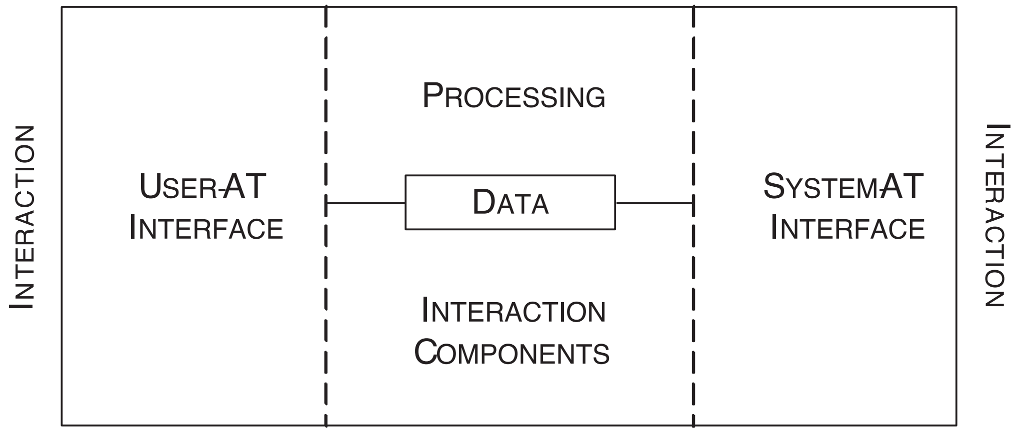


Figure 2.13: A model of assistive technology

In effect, an AT looks like two back-to-back systems, one of which interacts with a system and one of which (usually) interacts with a user. Where multiple ATs are connected in series, both interfaces of a given AT may be interacting only with other systems.

The AT interaction components provide the basic styles and types of media that can be used by its processing functionality to instantiate the two interfaces. This set of components is vital to how the given AT performs its transformation tasks. Each interaction style and/or media type may be used any number of times within the resulting interfaces. Note, the UARM does not assume that a user's interaction abilities and skills will match an AT's interaction abilities and skills. This implies that a user's interaction with a system may still be handicapped even in the presence of an AT.

When an AT receives input through one of its channels, it acts on the input according to its processing mode. ATs involve combinations and variations of two processing modes: pass-through and transformation.

Pass-through mode allows the content of a channel, to pass through the AT without modification. Since ATs typically specialize to a subset of all available channels, this is the mode that will influence all channels the AT is not specialized in.

Transformation mode involves the AT receiving input from a channel, taking its content and transforming it into a new medium. The process used for transformation can be fixed, user-selectable, or user-modifiable. This newly transformed content is then output to the client through a new channel. The input channel may then be closed. However, when combined with pass-through mode, the client will receive both channels. An AT may choose one or more output channels to use for a given input channel. This choice may be made automatically or under user control.

Many hardware-based ATs perform automatic transformation of channel content as appropriate to their hardwired behaviour. For example, a glare filter, an AT that reduces the amount of glare from environmental lighting that is reflected back to a user by a screen, has no intelligent features whatsoever.

User-controlled transformation allows the user to set the parameters or focus of any transformation. That is, a user may, through a set of transformation rules and/or preferences, influence how the AT is to perform the transformation of content. Such transformations are most easily performed by software-based or programmable ATs. For example, a screen reader with voice output may have several options that are set by its user including the voice gender, speech rate, and pitch.

In the UARM, the function of the user's interface is to create, select, and manage the combination of channels (through the user's physical senses) that are used to interact with systems and ATs. The user's interface provides management of the various channels to allow the transmission/reception of multiple messages at the same time, and allows the user to focus on particular channels (Carter & Fourney, 2004b).

2.3.2 AT and Accessibility

Accessibility is only possible when users and systems use compatible interfaces for communication. The inclusion of an AT allows translation between two incompatible interfaces as illustrated in Figure 2.14.

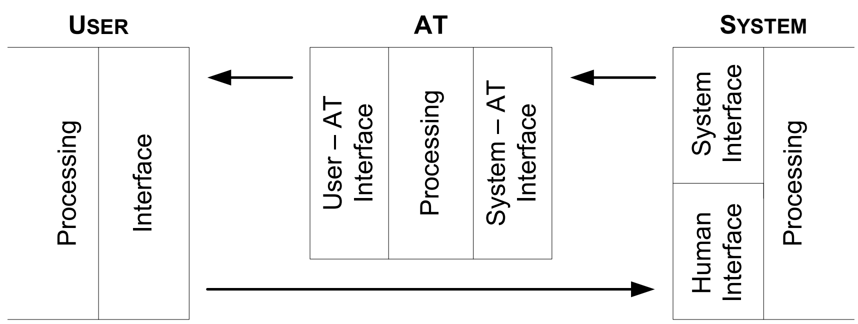


Figure 2.14: Using AT to help interface between components

Different situations call for different packagings of systems. Where the user's goal is to interact with a particular application package, the user may choose the operating system, computer, peripherals, and other ATs that make the application the most accessible. ATs may be required to achieve or to improve accessibility where the user's goal is to interact with an application package that is part of an existing hardware/software system.

The model presented in Figure 2.14 holds in all situations regardless of the different possible locations of system boundaries. In this model, ATs can be considered anything that is added to the basic system to make it accessible to users. Multiple ATs may be used in sequence and/or in parallel to support accessibility. As suggested by the UARM, the AT could interact with either the system's screen (i.e., user interface) or bypass it and interact directly with the system's functionality.

Figure 2.15 illustrates the paths between the User and their ultimate goal, the application (A1, A2, A3) the user wants to use. Multiple communications can occur in either direction along the connecting lines between components. The applications being used must be accessible to the user. To this end, Software-based Assistive Technology (SAT) might be needed.

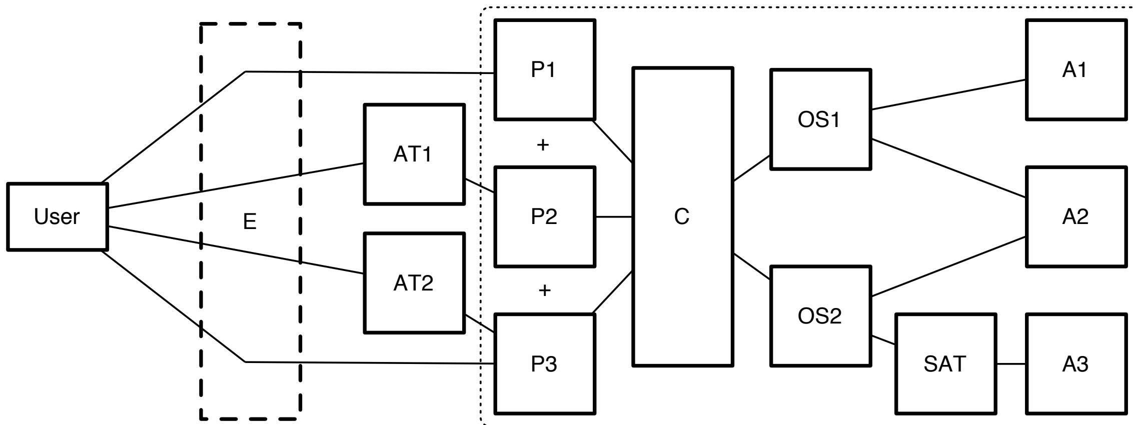


Figure 2.15: Components of accessibility

Although not traditionally considered AT, each of the layers (Operating System, Hardware, Peripherals, Assistive Technology, Environment) between the user and the application has the same effect as an AT in either increasing or decreasing accessibility. The choice of operating system (OS1, OS2) to use with the application can limit or increase the user's accessibility with the application. It may limit accessibility where it does not support certain forms of interaction between the user and the application. It may increase accessibility where it supports transformations of communications between the user and the application from one form of interaction to another. The computer (C) with which the operating system interacts may limit the user's experience still further. Users are also limited by the capabilities of peripherals (P1, P2, P3) available with the computer.

The user may perceive the combination of application, operating system, computer, and peripherals as a single system, as is indicated by a dotted box in the figure. When considering accessibility, these components may be modelled separately or as a single system.

Assistive Technologies (AT1, AT2) may be used to transform interactions of peripherals to make them more accessible. Software-based ATs (SAT) may also be present to transform interactions of applications. Environmental conditions (E) may further degrade the accessibility of certain interactions.

To the user, the total experience with all of these components might be perceived as a total system. It is the total system that must be specified to evaluate accessibility for the user.

2.3.3 Specifying a Total System

Communications are transmitted (by systems, users, or ATs) through channels (environments) to their intended receptors (systems, users, or ATs). Accessibility exists when the receptor is able to receive and understand the message as transmitted. Systems, users, ATs, and environments can be considered Interacting Components (IC). Individual communications can be modelled in terms of the receptors, channels, and transmitters used to accomplish the communication. Interaction involves many sets of communications going in either direction between the ICs in the interaction.

An accessibility framework modelling all of the sets of transmitters, channels, and receptors involved in the set of possible interactions between a particular user and a particular system can be used to evaluate the accessibility of a system in a given environment to a particular user.

This accessibility framework involves multiple sets of interactions each of which is composed of one or more sets of {*receptor*, *channel*, *transmitter*}. Rather than deal with each interaction, it is possible to model the set of potential interactions based on an understanding of the compatibility of transmitters, receptors, and channel characteristics of the ICs.

Since the communication possibilities and/or needs of a user at a given time may be greater than the user's capabilities, an AT information management function may optimize the set of channels being used to help the user avoid information overload. This management may include filtering and/or ignoring various competing channels or placing focus on specific channels. The CAP was designed to meet this need.

2.4 Common Accessibility Profile (CAP)

The idea of a CAP comes from the need to describe both user-system accessibility and user-AT-system accessibility across all users and systems. Being able to describe accessibility has several benefits such as the opportunity to evaluate and improve accessibility. CAPs describe the different communication skills or capabilities of an object. CAPs provide a mechanism from which a comparison of the skills of objects can begin. Comparing these skills shows the accessibility of a system for a user. The CAP describes potential classes of communication, which may involve multiple channels, providing the descriptors needed to evaluate an interaction. This section is based on material that the author developed which is currently being advanced by ISO/IEC JTC1/SC35, in a project led by the author, as the basis of a new international standard (International Organization for Standardization [ISO] and International Electrotechnical Commission [IEC], 2007a).

There are two main types of channels {*system to user*, *user to system*}. When an AT is used in the communication, it introduces at least two additional channel types. The AT takes the place of the system for the user {*AT to user*, *user to AT*} for some or all interactions. The user interacts with the AT (hopefully seamlessly) as if it is the full system or a regular part of the system. The

AT also takes the place of the user of the system $\{AT\ to\ system,\ system\ to\ AT\}$ for some or all interactions. The system interacts with the AT as if the AT is the user. Thus, the addition of AT to the UARM increases the number of channel types, the number of channels and the interfaces to them.

A CAP can be used to describe the transmitter and receptor characteristics of the ICs. A pair of CAPs can be used to identify the need for a channel connecting ICs. CAPs can be combined to evaluate the resulting accessibility combinations of specific users, specific systems, specific ATs, and specific environments. For a given IC, a CAP is specified in terms of:

$$\begin{aligned} CAP\ (Interacting\ Component) = \quad & \{\{Input\ Receptors\}, \\ & \{Processing\ Functions\}, \\ & \{Output\ Transmitters\}\} \end{aligned}$$

The CAP of each IC may involve one or more sets of specifications of Input Receptors (IR), Processing Functions (PF), and Output Transmitters (OT). Describing PFs is optional for users and systems, but is required for ATs, which translate between multiple channels. A set of specifications can include combinations of IR, PF, and OT.

ICs may make use of one or more OTs and/or IRs. Where multiple OTs or IRs are required they will be AND'ed, which is the default assumption. Where substitutions of OTs or IRs are possible, they will be OR'ed within the CAP specification. For example, (IR1 AND (IR2 OR IR3)) requires that input receptor IR1 always be used and that either input receptor IR2 or input receptor IR3 be used.

Systems are intended to help users perform tasks. Systems may or may not be directly accessible by users. The CAP of a system provides the starting point for evaluating and improving the accessibility of the system for a user in a given environment. The environment may reduce the accessibility of a system. ATs may be used to increase the accessibility of a system. Thus, an evaluation of accessibility involves analyzing the CAPs of a set of systems, users, environments, and ATs.

2.4.1 Describing Input Receptors, Output Transmitters, and Processing Functions

Since communication is only possible where there are corresponding IRs for the OTs being used, a common format is used to described both IRs and OTs. Both IRs and OTs can be specified as follows:

$$\begin{aligned} Input\ receptor\ /\ Output\ transmitter = \quad & \{interacting\ component, \\ & direction, \\ & modality, \\ & properties\} \end{aligned}$$

Processing transforms communications between inputs and outputs and is represented by a pair of input and output formats along with a rule to describe the transformation. User and system processing is usually outside the bounds of evaluation. AT processing effects the communication by transforming its characteristics. Processing can be specified as follows:

$$Processing = \{Input\ receptor,\ Transformation,\ Output\ transmitter\}$$

The remainder of this section describes these elements of IRs and OTs along with a running example.

2.4.1.1 Interacting Components

The CAP recognizes four basic types of ICs {*user*, *system*, *AT*, *environment*}

$$IC\ is\ one\ of:\ User[name],\ System[name],\ AT[name],\ Environment[name]$$

User, System, AT, or Environment provides a general identification of the IC. A specific name can be added to this general indication to identify a more specific IC. Although ATs are just special cases of systems, they are included in this list because it is useful to consider them as distinct (i.e., their use often may be optional).

The CAP of a given System IC could be provided by the component's manufacturer and is based on the design and specifications of the component.

The CAPs of all of the ICs in use can be stored locally in a local database of CAPs. This database may be augmented by a centralized Standard Database of CAPs from which locally stored CAPs may be updated.

The CAP of a user can be provided by the user through tools such as user preferences/settings. In this way, users can describe their capabilities. A user's habits/preferences for certain capabilities may also be described within a CAP either by only setting those capabilities that are preferred or through additional preference settings provided by the IC.

The CAP regards a software application in two ways: as a component of a system or as an integrated part of a system.

As shown in Figure 2.15 on page 39, in certain circumstances, a software application can be considered a separate IC that interacts with other ICs in the system:

- A CAP may be needed to describe the additional interaction styles and media that can be handled by the system when this software is available. For example, a software-based AT such as a screen magnifier provides new interaction styles to the existing system, which may not normally be part of the system's own interaction styles.

- A CAP may be needed to describe the loss of specific interaction styles and media that can no longer be handled by the system, when this software application is running. For example, if a given software application is known to break a specific accessibility feature, the system needs to know so that, whenever this application is running, it is "aware" that a limitation to its functionality has been added.

- Finally, occasions such as the upgrade of the system or installation of new software may necessitate adding the software application's CAP to the local database to describe new interaction styles and media that can be handled by the system.

As shown in Figure 2.12 on page 36, for the CAP a software application is usually an integrated part of a system. Since a CAP for a system describes the system, then usually the CAP for the system would include this information without the need for any additional CAPs.

For the purpose of a running example, a System IC named "Example" is used. The IC System[Example] is initially provided by the component's manufacturer.

2.4.1.2 Direction

Direction indicates whether a communication is an input to or output from an IC. A communication can begin at the user, system, or AT. In addition, communications can originate from the environment. Channels only exist where the user, system, and any ATs have compatible interfaces.

$$\begin{aligned} Direction = \quad & \{In, \\ & Out, \\ & Dual\ (\textit{applicable to environment only})\} \end{aligned}$$

For example, the interface is an output of the IC System[Example]. Its direction is "out".

2.4.1.3 Modality

Modality describes how the content of the communication is presented. The three basic modalities are:

$$\begin{aligned} Modality = \quad & \{Visual, \\ & Auditory, \\ & Tactile\} \end{aligned}$$

Content presented in a visual modality like text and graphics are perceived through sight by a user. Content presented in an auditory modality like speech and music are perceived through hearing by a user. Content presented in a tactile modality like pressure and heat are perceived through touch. Touch (i.e., "haptic") perception incorporates both kinaesthetic sensing (i.e., proprioception) of the position and movement of joints and limbs, and tactile sensing of a stimulus through the skin (e.g., heat, pressure) (Colwell, Petrie, Kornbrot, Hardwick, & Furner, 1998).

Where there is a mismatch between the modalities and/or properties of OTs and IRs, the accessibility of communications may be limited. ATs may be needed to translate between media and/or between interaction styles.

For example, the interface of System[Example] is output using a "visual" modality.

2.4.1.4 Properties

Properties describe the nature of the content of the message. The properties of a communication include the media, languages, and interaction styles contained in the communication. Additional properties, relevant to accessibility, may be identified. The set of properties can be described in terms of the media, languages, and interaction styles contained in the communication.

$$\begin{aligned} Properties = \quad & \{\{Media\}, \\ & \{Languages\}, \\ & \{Interaction\ Styles\}, \\ & \{additional\ properties\}\} \end{aligned}$$

Potential additional properties include future properties not currently envisioned. Such properties may be defined by a future International Standard. Other potential additional properties may include media formats or interaction styles that are experimental or based on a manufacturer's product but not yet accepted as an industry or international standard.

For the IC System[Example], each property will be discussed in turn.

Media

Media describes how the content of the communication is to be rendered. As shown in Table 2.1, media can be subdivided in terms of whether a communication is text-based or not text-based, whether it is static or dynamic, and the particular modality in which it is implemented.

Table 2.1: Subdivisions of "Media" (adapted from Fourney, 2004)

Modality	Text-based	Non text-based
Visual	written text	pictures graphs animations (dynamic only) movies (dynamic only)
Auditory	spoken text (dynamic only)	sounds (dynamic only) music (dynamic only)
Tactile	Braille gestures	textures tactile graphics force feedback temperature

Media, which should default to include all media, can identify a single medium or multiple specific media, where appropriate.

$$
\begin{aligned}
Media = \ & \{ALL, \\
& Written\ Text, Spoken\ Text, Braille, \\
& Picture, Graph, Animation, Movie, \\
& Sound, Music, \\
& Texture, Tactile\ Graphic, Force\ Feedback, Temperature, Gestures\}
\end{aligned}
$$

Where multiple media are required they will be AND'ed, which is the default assumption. Where substitutions of media are possible, they will be OR'ed within the Media property.

For example, the interface of System[Example] uses "picture" media (i.e., icons) to output information through a "visual" modality.

Languages

Most media, including non-textual media, involve certain cultural and/or linguistic dependencies. Text-based media, which can be either written, spoken, or signed, are provided in a specific language. Non-text based media might only be meaningful under certain cultural or linguistic constraints. The Language property is used to specify the languages and/or cultures required to make the communication accessible. Language is a required property for which there is no default value.

Communications may involve one or more languages. Where multiple languages are required they will be AND'ed, which is the default assumption. Where substitutions of languages are possible, they will be OR'ed within the Language property.

Although the interface of System[Example] is not text-based and may even be universally understandable regardless of culture of origin, a language setting is still required. For example, one might say that System[Example] presents content based on the English language.

Interaction Style

Interaction style describes the rendering of particular content objects within the selected medium. Interaction style, which should default to include all styles, can identify a single style or multiple specific styles, where appropriate.

$$
\begin{aligned}
Interaction\ Style = \ & \{ALL, \\
& Menu, Command\ Line, Direct\ Manipulation, Form\ Fill, \\
& Natural\ Language, Gesture\}
\end{aligned}
$$

Note that Gesture is included as a Media type, an Interaction Style, and – under "Tactile" – as a Modality. Gesture is not simply a modality for rendering interaction styles or content. Gesture can be the basis of a Natural Language interface (i.e., based on a signed language) or a Direct Manipulation interface (e.g., Tom Cruise in *Minority Report*).

However, gesture provides additional opportunities for interaction. For example Konami's *Police 911* (also known as *Police 24/7*) is an arcade game which uses motion detectors to sense body movement rather than requiring the player to move individual controls. This innovative system requires full-body movement and simulates many realistic tactical manoeuvres, such as dodging bullets and "taking cover" by ducking in front of the game console (see Appendix B for more information).

For example, if the interface of System[Example] uses "picture" media (i.e., icons) to output information through a "visual" modality, then one interaction style it might support is "direct manipulation". Although this running example may seem trivial, it illustrates the descriptive potential of the CAP model.

Other Properties

Other properties can be added to the CAP to support accessibility. Descriptions of any other properties added to a CAP specification should be made available to the users of the specification.

All properties or features of an IC could be referred to as "Interacting Component Features". To reduce confusion between this phrase and "Interacting Component", it is useful to shorten the phrase. Thus, this Thesis will use the phrase "Component Features" (CF) to refer to features and properties of ICs.

2.4.2 Using Channels in the CAP

Channels connect CFs (IRs and OTs). Section 2.4.1 discussed the elements of IRs and OTs. This section will discuss how CFs use channels. First, this section illustrates, through a series of examples, how channels are used by CFs. The remainder of this section outlines the features of channels and how their use can be described in the CAP.

Note that this section emphasizes the logical nature of channels as connections that transmit communications. Channels are assumed to exist wherever a user and a system have a corresponding pair of an IR and an OT. This section assumes that channels are not documented directly in the CAP. Channels are not documented unless limitations in a channel cause it to be documented as an environment.

This assumption is a simplification that works in the context of this section and aids in the understanding of how channels are used in the CAP by users and systems. As mentioned above, a CAP can be used to describe the transmitter, channel, and receptor characteristics of the ICs.

2.4.2.1 Considering the Need for Channels

Figure 2.16 shows a system that can be fully accessed by the user. There are two ICs in this figure, a user and a system.

The user IC, depicted on the left, consists of three CFs: an OT for the auditory modality, an IR for the visual modality, and an OT for the tactile modality. These CFs represent the user's speech, vision, and kinesthetic capabilities respectively.

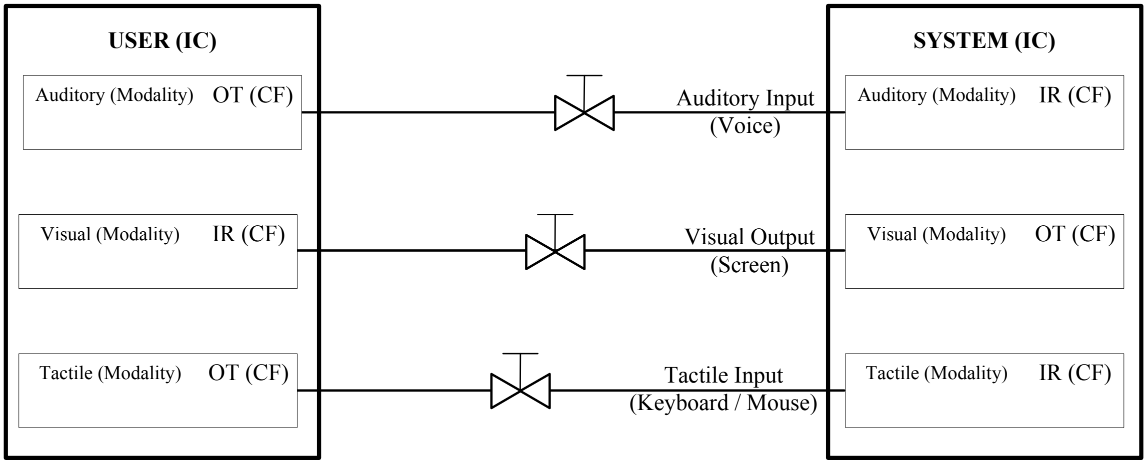


Figure 2.16: A fully accessible system

The system IC, depicted on the right, consists of three CFs: an IR for the auditory modality, an OT for the visual modality, and an IR for the tactile modality. These CFs represent respectively the system's auditory input capabilities (e.g., a microphone), visual output capabilities (i.e., the screen), and tactile input capabilities (i.e., the mouse and keyboard). These capabilities represent the system's interface and, as such, provide a set of expectations on the part of the system of the user's capabilities.

Connecting each IR to a companion OT is a channel. Note that the valve used in the UARM is shown here to represent the potential for handicaps to the interaction. Recall that channels are one-way communications from output (OT) to input (IR).

Thus, Figure 2.16 shows that the user is capable of interacting with all system modalities. When users are able to interact with all system modalities, the system can be considered to be accessible. Note that while system capabilities are matched by user capabilities, if a user has any other capabilities (e.g., sense of smell), it does not need to be matched to the system.

Figure 2.17 shows a system that cannot be fully accessed by the user. As with Figure 2.16, the user, depicted on the left, consists of three CFs: an OT for the auditory modality, an IR for the visual modality, and an OT for the tactile modality. These CFs represent the user's speech, vision, and kinesthetic capabilities, respectively.

However, the system, depicted on the right, now consists of four CFs: an IR for the auditory modality, an OT for the visual modality, an IR for the tactile modality, and a new OT for the auditory modality. These CFs represent the system's auditory input capabilities (e.g., a microphone), visual output capabilities (i.e., the screen), tactile input capabilities (i.e., the mouse and keyboard), and auditory output capabilities (e.g., a speaker), respectively.

While each of the user's IRs/OTs have connections to their respective system OTs/IRs, the system's auditory modality OT does not have a corresponding user IR. When system capabilities

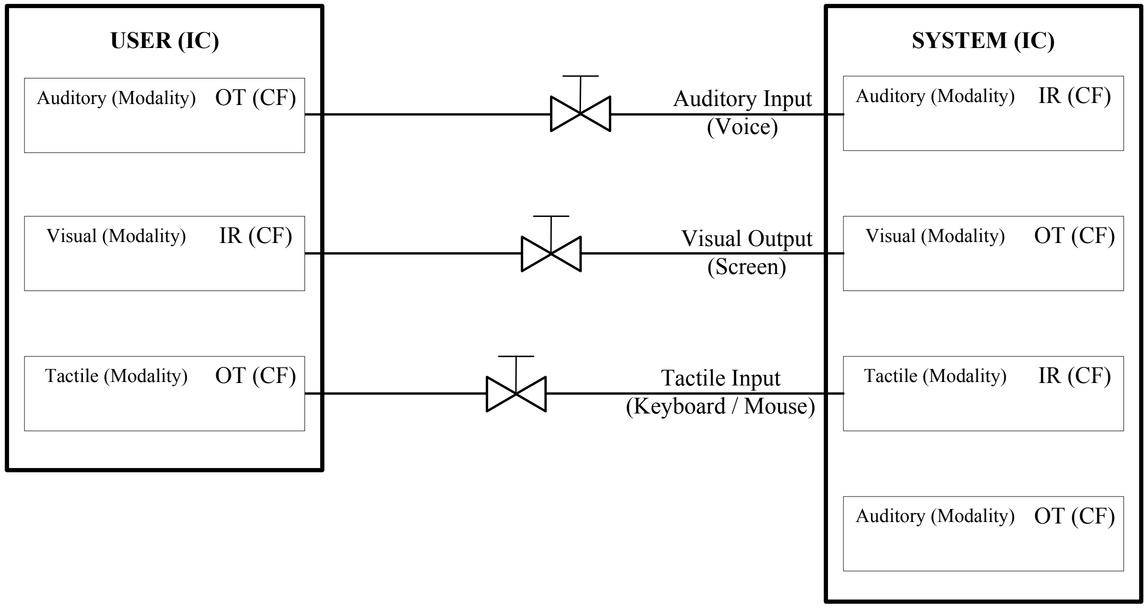


Figure 2.17: A system with access issues for the user

like this are not matched by an equivalent user capability, access issues exist. Such access issues might be resolved to some degree by the addition of appropriate AT transforming the system's auditory output into visual output that the user can use. Users cannot add new capabilities in this same manner.

Thus, Figure 2.17 shows that the user is capable of interacting with most system modalities. When users are not able to interact with all required system modalities, the system can be considered to be inaccessible.

Figure 2.18 corrects a fallacy that was shown in Figure 2.16 on page 47. While Figure 2.18 still shows a system that can be fully accessed by the user, it depicts that the mouse and keyboard are separate tactile modality IRs for the system. However, for the user, the mouse and keyboard use the same tactile capability.

The mouse and the keyboard are each connected to a single system tactile input channel. Both of these channels are connected to the user's tactile output capabilities. The user's OT is able to use multiple channels. The system's IRs do not appear to support multiple channels. Further, if the user types with one hand and uses the mouse with the other, the user can use these channels at the same time.

In addition, the user's tactile output capability can be fully used without stopping. The user can utilize the mouse and/or keyboard in sustained tactile communication. This can be extended to the user's auditory modality output CF in that the user can, through their voice, provide sustained continuous auditory communication to the system.

This example suggests several issues to consider when documenting channels:

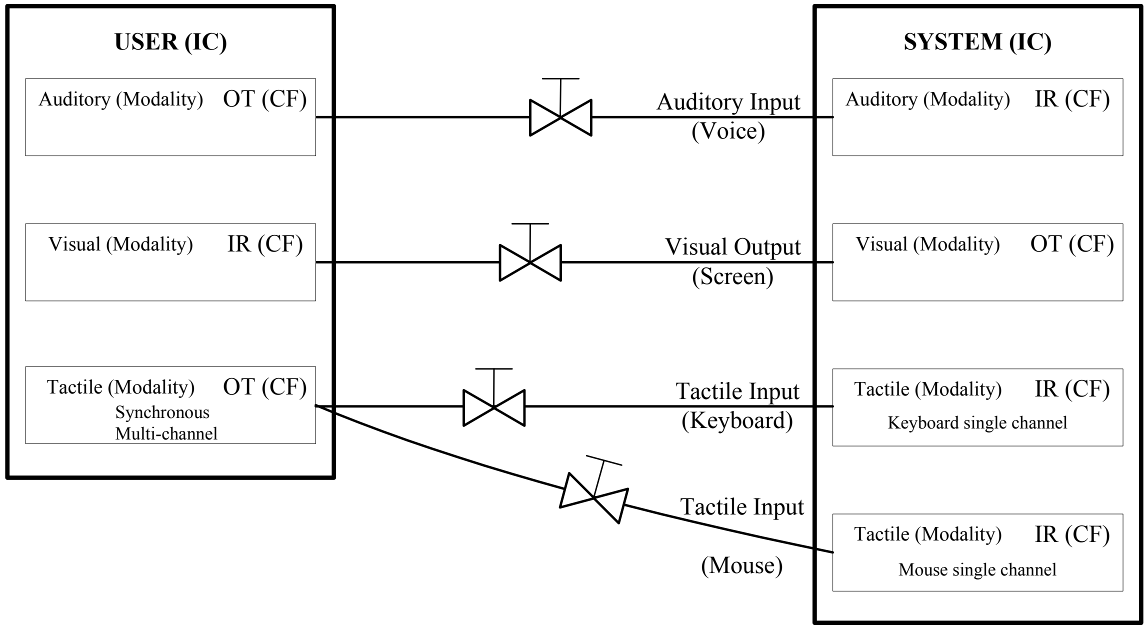


Figure 2.18: Multiple channels not interfering with each other

- The CFs of a system might only be able to accept single channel input.
- The CFs of a user can produce multi-channel output.
- The CFs of a user can transmit sustained continuous communication.
- The CFs of a system can receive sustained continuous communication.

As shown in Figure 2.19, systems may not always be able to use multiple input channels to the same IR. A new IC has been added to represent the environment. Any object in the environment can potentially be represented as an IR or OT of that environment.

In this environment, there is a radio playing. The radio is represented by an auditory modality OT. Since the user does not have an auditory modality IR, they are not listening to the radio (and may not even be aware that it is playing). However, the system does have an auditory modality IR. This IR is responsible for receiving voice input. The presence of the radio is adding further information, or noise, into this process.

Thus, this CAP shows an auditory channel conflict because there is insufficient channel capacity on the part of the system. Consequently the user is no longer able to employ their auditory output capabilities (i.e., voice) to interact with the system.

This example suggests several issues to consider when documenting channels:

- The CFs of a system might be able to accept multiple channel input. However, there might also be conflicts among the various inputs of information sent to one CF. To avoid overload, a system's CFs might be forced to accept single channels only.

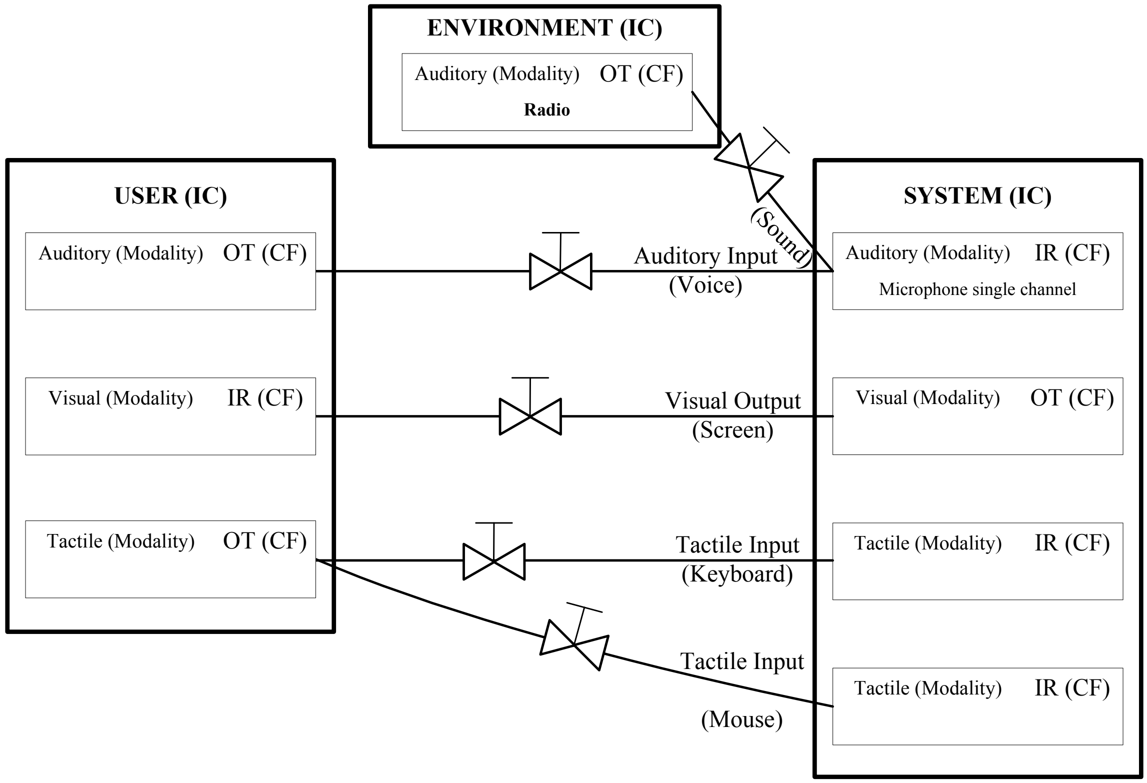


Figure 2.19: Auditory channel conflict (insufficient channel capacity)

- The CFs of an environment can transmit sustained continuous communication.

Figure 2.20 shows that the auditory channel conflict from Figure 2.19 may be resolved by adding another auditory input IR to the system, such as (for example) a boom microphone. The addition of this microphone can enhance the system's ability to filter the user's voice from the environmental noise. Also, the existing microphone sensing the environment could still be used by the system to detect specific environmental noises (e.g., a fire alarm) and then notify the user of the sound.

This example suggests several issues to consider when documenting channels:

- The system has limited channel capacity, based on the characteristics of its IR's and OT's.
- Increasing the channel capacity of the system, by adding or replacing IR's and/or OT's, can resolve certain channel conflicts.

Figure 2.21 shows a more complex system that can be fully accessed by the user. There are three ICs in this figure, a user, a system, and an environment.

The environment consists of one CF, namely an OT for the auditory modality. This CF documents a radio.

The user consists of four CFs: an OT for the auditory modality, an IR for the auditory modality, an IR for the visual modality, and an OT for the tactile modality. These CFs represent the user's

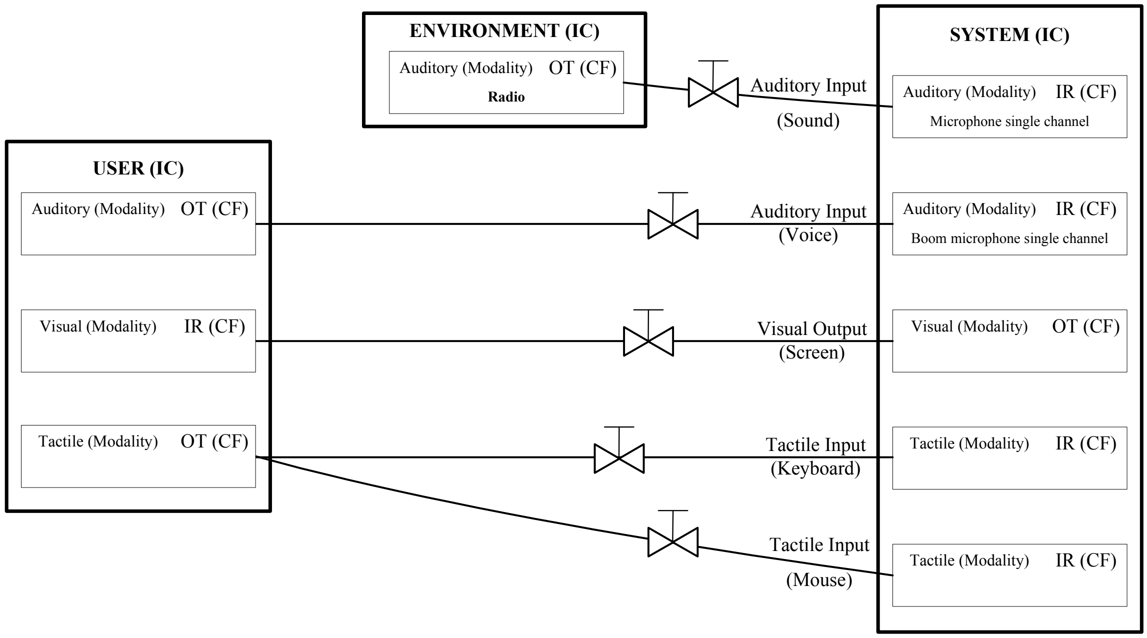


Figure 2.20: Adding a boom microphone auditory modality to the system

speech, hearing, vision, and kinesthetic capabilities, respectively.

The system consists of seven CFs: two IRs for the auditory modality, two OTs for the auditory modality, an OT for the visual modality, and two IRs for the tactile modality. These CFs represent the system's auditory input capabilities (e.g., a microphone, a boom microphone), auditory output capabilities (e.g., a music player, system bell), visual output capabilities (i.e., the screen), and tactile input capabilities (i.e., the mouse and keyboard), respectively.

Where Figure 2.18 depicts the ability of the user to manipulate the mouse and keyboard via a single channel, this CAP depicts the capability of the user to receive a variety of auditory input via multiple channels. While the music player provides a continuous auditory output, the system bell is intermittent. This means that while both channels are in use between the system and the user, there is no auditory channel conflict (i.e., cognitive overload) on the part of the user.

Cognitive load is the level of effort associated with thinking and reasoning (including perception, memory, language, etc.), thus potentially interfering with other thought processes (Sweller, 1988). To avoid any cognitive overload, a user interface needs to minimize the cognitive load associated with using the interface itself so that all of a person's cognitive resources are available for their task (Foraker Design, 2005).

This example suggests several issues to consider when documenting channels:

- CFs can output either intermittently or continuously.
- The potential exists for cognitive overload of the user (e.g., auditory channel conflict).

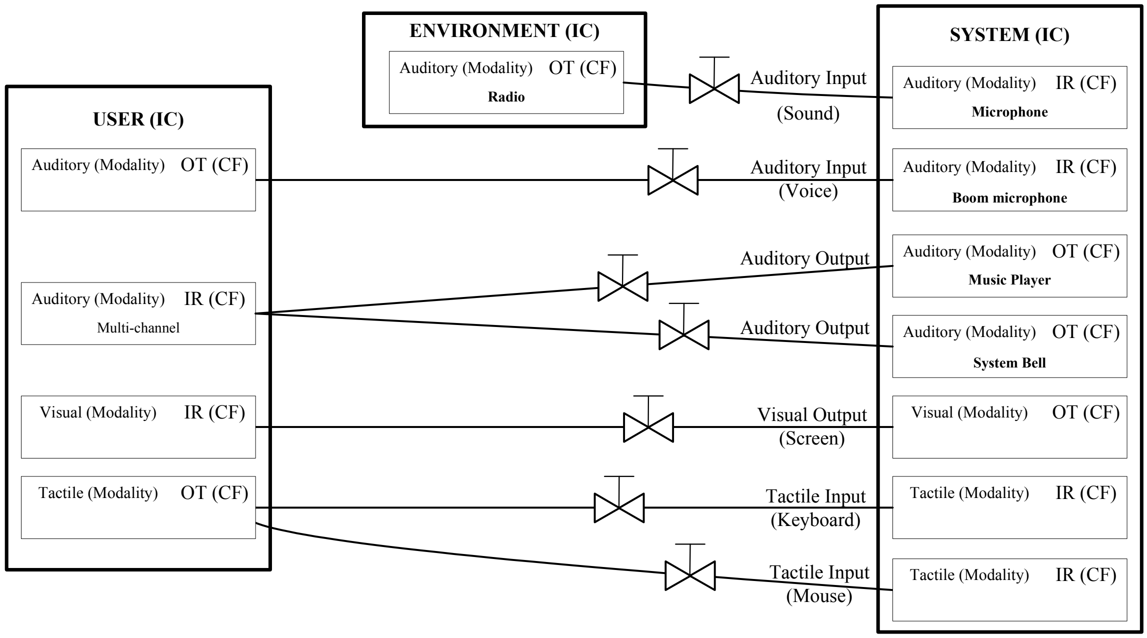


Figure 2.21: User capabilities for multiple input channels

2.4.2.2 Properties of Channels and the CFs they Connect

The purpose of this subsection is to outline the properties of channels of interest to the CAP and how they can be represented within CAPs. This discussion is based on the examples shown above.

By considering channels as connections and focusing on the CFs they connect, a problem arises if both systems and users are allowed to connect to any number of channels and if any channel is allowed to connect to any number of systems and users. This would result in an N:N mapping problem. The CAP needs to recognize the existence of multiple connections between channels and users, requiring that the user side of this mapping remain an N. However, the N:N problem can be handled by placing restrictions on the modelling of CFs of systems. By defining system CFs as simple CFs capable of connecting to only a single channel, it is easier to see how they relate to users. Multi-channel capabilities can be described by logical combinations of these simple system CFs. This allows reducing the connection problem to a series of N:1 connections between a user and a system.

The above examples suggest several concerns regarding channels:

1. The CFs of a system have limited channel capacity.

2. Increasing the channel capacity of the system by adding additional CFs can resolve certain channel conflicts.

3. To avoid system overload, CFs of a system are best conceptualized to be able to accept only single channel input. Where they can accept multi-channel input, they may become confused

regarding the validity of the input.

4. The CFs of a system are best conceptualized as transmitting only single-channel output. Where multiple-channel output is created, it can be considered as a multiple of single channel output and thus described in terms of multiple CFs.

5. The CFs of a user can produce multi-channel output.

6. The CFs of a user can receive multi-channel input.

7. The potential exists for documenting a user's cognitive load by noting the potential for channel conflicts.

8. The CFs of a user or system can transmit either continuous or intermittent communication.

9. The CFs of a user or system can receive intermittent communication. While not shown in the examples above, it is reasonable to expect that CFs of systems can also receive intermittent communication.

10. The CFs of a user or system can receive continuous communication.

11. The CFs of an environment can transmit sustained continuous communication. While not shown in the examples above, it is reasonable to expect that CFs of environments can also transmit intermittent communication.

These concerns suggest several common characteristics that can be used to describe channel connections between CFs.

First, a CF can be connected to one or more channels. If a CF can only be connected to a single channel, it can only accept one connection. If a CF can be connected to more than one channel, it can accept any number of connections. Since a CF has a conceptual rather than physical relationship to the IC that it represents, it is easier to identify and use specialized system CFs that are only connected to a single channel. This simplification, however, is not appropriate in dealing with users who must be capable of dealing with multi-channel interactions.

While the Zero-One-Infinity Rule (MacLennan, 1983) does apply to the use of channels, the absence, presence, and number of channels has different effects. As mentioned earlier, the absence of a channel means that two ICs (e.g., a user and a system) cannot interact with each other using a specific modality. The presence of a single channel means that two ICs can interact with each other using a specific modality. The presence of multiple channels requires the user to have the cognitive capacity to manage multiple channels in the same modality. With increasing numbers of channels, comes an increasing potential for cognitive overload.

Second, a CF can use a channel intermittently or continuously. If the CF only uses a channel intermittently, it can interleaf multiple inputs or outputs and work with several communications.

In the examples above, the system bell could share the user's auditory IR along with another intermittent auditory output because the system bell only needs the IR intermittently. If an IR/OT communicates continuously to an OT/IR, it cannot share that channel with other IRs/OTs since, if there is only one channel, it cannot make that channel available.

In the case of users, there may be cognitive overload issues when receiving continuous communication with a single IR (within a single modality). For example, a user may not be able to receive sustained auditory input for long periods of time without a break. Such sustained input may lead to user fatigue and/or adaptation to the stimulus.

Third, as already noted above, a CF can either communicate over one channel or over several channels. This specific property of a CF implies that, if it is able to use multiple channels at the same time, a CF may or may not be capable of concurrent communication.

On the one hand, the CF may be capable of concurrent inputs (or outputs) on more than one channel at a time. That is, the CF might be able to receive (or transmit) communication via one channel while also receiving (or transmitting) communication on another channel. Since the CF receives (or transmits) communication on behalf of the IC, it is possible that this information management is performed by the CF. However, knowing that a CF has this property might allow the IC to assume the role of managing this information.

On the other hand, some CFs might need to use multiple channels one communication at a time. That is, the CF might be able to receive (or transmit) communication via one channel only after having completed receiving (or transmitting) communication on another channel. It is possible that this information management is performed by the CF. However, if an IC knows that a CF has this property then the IC might be able to perform this information management itself.

If the CF is only able to use some or all of its channels serially, it is interruptible and consequently might only be able to handle inputs or outputs in bursts. Note that if a CF is only capable of using a channel intermittently, it is very likely that it will only input (or output) communications serially.

Fourth, systems have limited channel capacity. While a system's channel capacity may be increased to resolve channel conflicts and to avoid system overload, users cannot add capacity in the same way. Documenting the channel capacity of a CF may help to relieve such an overload.

Fifth, since not all communications are sustained and continuous (i.e., some are intermittent), then some channels are more exclusively dedicated to the use of a specific CF than others. Those channels which are being intermittently used may be available for use by more than one CF of the same modality or may be exclusively for the use of one CF.

Since communication occurs between corresponding IRs and OTs in one direction, we can further describe IRs and OTs in terms of their use of their communication capabilities.

$$
\begin{aligned}
Communication\ capabilities = \ &\{Capacity,\\
&Parallel,\\
&Serial,\\
&Degree,\\
&Intermittent,\\
&Devoted\}
\end{aligned}
$$

Where:

Capacity is the maximum number of channels the CF could accept. Since some CFs can only accept one and only one channel (e.g., system IRs), this channel feature allows documenting such restrictions. Note that this property differs from Degree, a fluctuating count of the current number of channels, and focuses on the limit of the maximum number of channels the CF can support. Among other issues, this property in combination with Degree allows the documentation of available capacity to determine whether the CF is able to accept new channels in a specific modality.

Parallel describes whether the CF can use multiple channels concurrently and, if so, how many channels it can use at the same time.

Serial describes whether the CF needs to use multiple channels successively and, if so, how many channels are needed.

Degree is the current number of existing channels connected to the CF.

Intermittent describes how many channels the CF is currently using for intermittent communication. All other channels are continuous. While *Intermittent* suggests the amount of time a CF may need to use a channel, it also describes how many such channels the CF will need. This is especially important for CFs that require continuous use of multiple channels.

Devoted describes whether the intermittent channels of the CF are exclusively for its use. If there are channels available that can be "borrowed" by other CFs then these can be made temporarily available for their use. CFs would share these available intermittent channels with CFs that use the same modality. Continuous channels are, by definition, devoted. If a CF uses a channel continuously, the channel is dedicated to the use of that CF until it is no longer needed.

An advantage to this approach of describing the channel capabilities of a CF is that it allows a CF to use multiple channels in different ways (e.g., a single CF could have both an intermittent channel and a continuous channel).

One disadvantage to this approach is that it does not document each channel individually. However, documenting how a CF uses each individual channel is not necessary, since it is sufficient to understand that a CF uses some number of channels in a specific way.

Another disadvantage to this approach is that it assumes that all channels of a CF (or shared between multiple CFs) have equal priority. Priority really matters when channels are shared. For example, if two CFs need to use the same channel, then it may be useful for one CF to have priority over the other due to the nature of its content. The current approach suggests that the same two CFs would have equal access to the same channel mediated in a manner similar to a process scheduler.

However, because of the *Devoted* property, a priority property is probably not needed. If the function of the CF is so important that it may need a channel, then it should have a dedicated channel even if that channel is intermittently used. Thus, we do not need to have CFs with priority over other CFs. Sharing channels only occurs where a particular function of a CF can wait (blocking) until resources (i.e., a channel) are released.

Thus far we have described both the properties of IRs, OTs and PFs as well as the ways CFs use channels. The next two sections will discuss how to apply CAPs.

2.4.3 Applying the CAP

CAPs can be used to evaluate handicaps to interactions, select ATs, and manage the use of ATs by systems. To apply a CAP, one must first acquire it. A database of CAPs for systems and ATs can be developed from existing technical specifications. However, a common format or coding method for CAPs is needed to realize such an application. In addition, tools to help create and manage CAPs are also needed. Such applications can then be further enhanced over time by adding user feedback. User feedback such as consumer ratings and/or reviews can differentiate solutions, assesses their quality, and evaluate product appropriateness from a consumer perspective (Carter & Fourney, 2004a).

Such user feedback, even if sparse or incomplete in nature, can identify the need for new AT products as well as provide feedback to existing AT products. The development of CAPs for users and environments can be performed through tools that work with the CAP database. Such tools can also help users to select the most appropriate ATs to improve their accessibility.

2.4.4 Applying the CAP to Identifying Handicaps

An algorithmic approach could be applied to the use of the CAP, when identifying handicaps to interactions, as shown in Figure 2.22.

The starting point is an identified user and an identified system. The identified system may be a composite of systems. A user's tasks (1) lead to the selection of specific operating systems

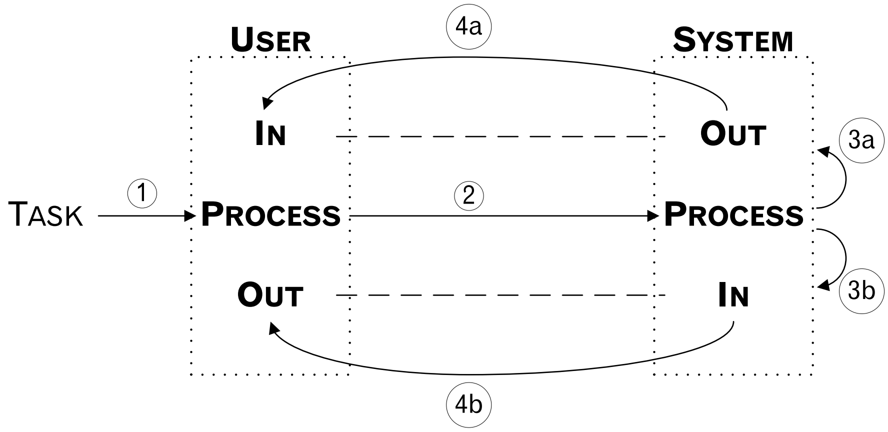


Figure 2.22: Identifying handicaps based on the CAP

and/or application programs to be used. Where particular selections are not made, this defaults to the selection of all known systems, operating systems, and/or applications. These processing requirements are compared (2) to system processing capabilities to select the channels needed for interacting with the user. This involves identifying channels (3a) to send information to the user and identifying channels (3b) to receive user communication.

Once these channel selections have been made on the system side, there remains a need to check whether or not the user has the ability to use these channels (4a, 4b). In situations involving multiple users, considerations must be made for each user.

If there are no alternative channels available to the user, handicaps exist. Further handicaps can be identified by the addition of environmental noise CAPs. These CAPs may add environmental handicaps to channels.

All of the resulting handicaps are opportunities for ATs. By starting with user and system CAPs, one can identify user needs that are handicapped.

2.4.4.1 Applying the CAP to Selecting ATs

An algorithmic approach to the minimization of handicaps to interactions can be developed given the CAPs of the user, system, and environment; the CAPs of potential ATs; and information on handicaps (in terms of channels).

The set of handicaps to interactions previously identified, can be used to find an AT that best minimizes as many of these handicaps as possible and that interfaces with the system, the user, and the environment. For an AT to interface with the system, it needs to be compatible to the systems' properties (media, styles, operating systems and/or applications). To interface with the user, the AT should use capabilities that the user has (e.g., literacy). To interface successfully, the environment should not excessively handicap the accessibility of selected channels (e.g., noisy environments can handicap speech output).

Once an AT has been selected, its CAP can be added to the CAP of the system. However, the addition of an AT may introduce new handicaps to the interaction (e.g., the best choice may require a capability the user does not have) that may require additional ATs. Thus, the process of identifying handicaps should be repeated after an AT has been added. If there are any handicaps to interactions remaining, interaction handicap minimization can be repeated until access has reached an acceptable level.

CAPs can also be used to help select combinations of systems and ATs for use in multi-user settings, such as those provided by educational institutions.

2.4.4.2 Applying the CAP to Managing ATs

CAPs can be used within a system's context to provide the system with information about how best to interface with its user and any ATs being used within the current environment.

CAPs can be created for different base and alternate configurations of users, environments, systems and ATs. These configuration CAPs can be entered directly into a system or created externally and then downloaded into these systems and/or accessed by these systems when required. They can be further loaded into ATs or accessed by ATs in situations where ATs are programmable.

Developing a Base Configuration

An initial base configuration of a situation can be described in terms of the set of CAPs representing all relevant systems, ATs, users, applications and environments. This requires the identification of the basics that are needed for accessibility, any optimal connections between multiple ATs, and any user preferences to add.

The base configuration must include the most optimal connections among all the ATs. Optimal connections reduce handicaps to the interactions. To identify these connections, the CAPs of each AT will need to be compared to ensure that no two ATs contradict each other. AT contradiction occurs when one AT blocks a channel used, in transformation mode, by another AT. Such contradictions might increase rather than reduce handicaps.

User preferences may be realized through either AT programmable settings or AT-system configurations. The CAP of an AT can include user preferences for different channels. This information may assist in the optimal configuration of an AT.

A base configuration CAP can be developed by selecting CAPs for appropriate system(s) and AT(s) from a Standard Database of CAPs and then entering CAPs for the intended user(s) and environment(s). A base configuration may be produced during the selection of ATs or later once the ATs have been procured and installed.

Developing Alternate Configurations

Once a base configuration has been established, it is possible to develop alternate configurations that respond to changes based on the needs of one or more specific users, applications, and/or environments.

Differences between users and even within a single user over time provide different accessibility needs. This is especially important in educational settings where multiple students with different accessibility needs may make use of a limited number of systems and ATs in an accessible lab. Different users have different abilities, skills, and/or preferences as well as different task needs when using the same system at different times (Carter & Fourney, 2004b). As a result, alternate configurations may be developed for multiple users to share the same system. This allows users who need different AT configurations (including configurations without ATs) to use the system. Alternate configurations may be developed either proactively or in response to such short-term changes.

Users of a system may utilize specialized CAPs when switching among various applications according to the task they are performing. Alternate CAP configurations may be developed for different applications which make use of different media/styles with differing levels of usability. The use of specialized application-specific CAPs can increase the accessibility of each individual application.

The physical environment within which the user is using the system may not remain stable. Alternate CAP configurations may be developed either proactively or in response to different environments. This allows the system to have ongoing awareness of its environment. The accessibility of mobile computing could benefit from the application of alternate CAP configurations that respond to changes in the environment.

The CAP for a new alternate configuration should start as a copy of the base configuration CAP. This new alternate configuration CAP can then be modified in one or more of the following manners: including additional (system / AT) CAPs from the Standard Database of CAPs, entering new and/or modifying existing (user / environment) CAPs, and/or deleting (system, AT, user, and/or environment) CAPs that do not apply to the new alternate configuration. There is also the need to be able to delete alternate configurations which no longer apply.

Reconfiguring Configurations

Changes to one or more of the system, AT, user, applications, and/or environment may result in the need to reconfigure the base and/or alternate CAP configurations. Changes can result from upgrades or replacements to the existing system (including additions or changes to the set of applications), the addition of new ATs and/or upgrades or replacements to existing ATs, changes in the regular environment, and/or changes to the user(s).

Differences between users and even within a single user over time provide different accessibility needs. Progressive or permanent changes to one's capabilities and/or preferences mean that each individual user may have different accessibility needs at different times. Such progressive or permanent changes to the user will involve a need to reconfigure their CAP configuration.

Educational institutions are likely to have the need to reconfigure existing configurations based on changes to the user population at least once per term or semester. Ensuring that a system can be changed when needed and can continue to meet the needs of multiple students will achieve the goal of serving the most students with the fewest resources.

Permanent modifications will change existing base and/or alternate CAPs. Temporary modifications can be accomplished by creating new alternate configurations.

2.4.5 Tying It All Together

The CAP provides a basis for identifying and dealing with accessibility issues in a standardized manner.

The CAP focuses on the CAPabilities of users, systems, and ATs and the minimizations of handiCAPs to interactions from the environment and incompatibilities among users, systems, and ATs.

CAPs can readily be developed for existing systems and ATs and can be custom developed for specific users and environments.

CAPs can be applied to identifying handicaps and to selecting and managing the use of ATs for individual users and for select groups of specific users.

The CAP can be used to help select and support combinations of systems and ATs for use in multi-user settings, such as those provided by educational institutions.

However, CAPs have several limitations. The CAP needs a common format or coding method to be computed. When systems are combined, so are CAPs; however there is no way to measure the effect of such a combination nor are there any algorithms to define such a combination. This Thesis will address many of these limitations.

2.5 Next Step

A model-based approach to evaluating and improving accessibility is needed. Since handicaps occur within interactions between users and systems, it makes sense to focus attention on interactions. The UARM, a standards-based usability model, uses the notion of interactions to describe accessibility. The UARM can be used to describe interactions both between users and systems and among systems combined in series allowing a convenient way to model the presence and use of ATs.

The CAP, which is based on the UARM, provides a framework of descriptors to describe user-system and user-AT-system accessibility across all users and systems. This framework can be used as a starting point to define metrics and algorithms that can be used to evaluate and improve accessibility.

Chapter 3

CAP Structure and Specification

As described in Chapter 2, Common Accessibility Profiles (CAPs) can be used to describe and evaluate accessibility between users and systems within particular environments. An overall CAP (CAP_O) is composed of the CAP_{IC} of each different IC, including those of: users (CAP_{USE}), systems (CAP_{SYS}), environments (CAP_{ENV}), and assistive technologies (CAP_{AT}).

$$\begin{aligned}(CAP_O) = \sum(CAP_{IC}) = \; & \textit{any } (CAP_{USE}) + \\ & \textit{any } (CAP_{SYS}) + \\ & \textit{any } (CAP_{ENV}) + \\ & \textit{any } (CAP_{AT}) + \\ & \textit{any other } (CAP_{IC})\end{aligned}$$

The CAP_{IC} of each IC is in turn composed of the CAP(s) of each Component Feature (CAP_{CF}), these include: the CAP_{IR} of each IR, the CAP_{OT} of each OT, and the CAP_{PF} of each PF involved in the IC.

$$\begin{aligned}(CAP_{IC}) = \sum(CAP_{CF}) = \; & \textit{any } (CAP_{IR}) + \\ & \textit{any } (CAP_{OT}) + \\ & \textit{any } (CAP_{PF})\end{aligned}$$

This chapter describes the information that is contained in each type of CAP. The next chapter will describe how different types of CAPs can be combined with one another to provide a CAP_{IC}.

3.1 CAP Structure

Table 3.1 outlines the structure of a high level CAP_O specification. Every CAP_O specification has an Identification Information section containing information such as the name of the CAP_O and its type (i.e., CAP_O). It may also contain an unstructured narrative description. Narrative descriptions can be used to record preliminary information and/or provide an easy to read introduction to the structured details of all CAP specifications. All useful CAP_O specifications have linkages to one or more CAP_{IC}(s) and may have linkages to other CAP_O(s).

Table 3.2 outlines the structure of a CAP_{IC}. Every CAP_{IC} specification has an Identification Information section with the unique name of the CAP_{IC} and the type (i.e., CAP_{USE}, CAP_{SYS},

Table 3.1: High Level CAP_O Structure

CAP_O Identification
Name
Type [CAP_O]
Description
CAP_O Linkages
Peer-CAP_O
Lower-CAP_{IC}

CAP_{AT}, CAP_{ENV}) of the IC specification as well as an unstructured narrative description. All CAP_{IC} specifications have linkages to one or more IR/OT/PF CF specifications as well as to the CAP_O(s) to whom it belongs.

Table 3.2: Interacting Component CAP_{IC} Structure

CAP_{IC} Identification
Name
Type [CAP_{USE} | CAP_{SYS} | CAP_{AT} | CAP_{ENV}]
Description
CAP_{IC} Linkages
Higher-CAP_O
Peer-CAP_{IC}
Lower-CAP_{IR}
Lower-CAP_{PF}
Lower-CAP_{OT}

Table 3.3 outlines the structure of a CF specification (CAP_{CF}). The CAP_{CF} structure of information is similar to the structure of requirements specified in the PUF methodology (Carter, Liu, Schneider, & Fourney, 2005; Carter, 2004). Every CF specification has an Identification Information section containing information such as the unique name of the CF and the type (i.e., CAP_{IR}, CAP_{OT}, or CAP_{PF}) of the CAP_{CF} specification. It may also contain an unstructured narrative description. Each CAP_{CF} specification has a linkage to the CAP_{IC} to which it belongs and may also have linkages to the CAP_{CF} of the other CFs to which this CF is directly connected within this IC (e.g., to other IRs/OTs and, as appropriate, to zero, one or more PFs). Linkages to other ICs are handled at the CAP_{IC} level. Other CF-specific information such as the modality and language of the message content are also included within the CAP_{CF} structure.

The remainder of this chapter will more fully describe the details of each type of CAP specification. Each subsection briefly describes the characteristic, or sub-characteristic, of the CAP being discussed and provides a breakdown, in table form, of the characteristic. A *Description* is provided to describe the intended function of the characteristic. *Possible values* for each characteristic are also provided.

Table 3.3: IC Component Feature CAP_{CF} Structure

CAP_{CF} Identification
Name
Type [CAP_{IR} | CAP_{PF} | CAP_{OT}]
Description
CAP_{CF} Linkages
Higher-CAP_{IC}
Peer-CAP_{IR}
Peer-CAP_{PF}
Peer-CAP_{OT}
CAP_{CF}-Specific Information
CF Type-Specific Information
IR/OT Type-Specific Information
PF Type-Specific Information

3.1.1 Identification Information

The term "Identification Information" is used here to encapsulate descriptive information that identifies and uniquely describes the object being recorded.

All CAP specifications share a common set of identification information. Table 3.4 shows that the Identification Information of a CAP specification states the type of CAP specification, the name of the CAP object being specified, and a human-understandable unstructured narrative description of the object.

Table 3.4: CAP General Format

	Description	*Possible Values*
Identification		
Type	The type of record. Dependent on specification type.	
Name	A unique descriptive name.	`any` (must be unique within the CAP)
Description	A narrative description to record preliminary information and / or optional comments further describing the object.	`any`

Type, identifies the kind of record being specified. Note that CAP_{IC} and CAP_{CF} are general templates used for multiple specific types of CAP(s) and are not appropriate to be used as a *Type*. Each of these types will be discussed below.

Name is some unique identifier for the object being specified. This name should be unique in the CAP and meaningful to the users of the CAP.

The *Description* field may optionally contain a narrative description to record preliminary information and/or provide an easy to read introduction to the structured details of the specification as well as comments that further describe or document the object. Descriptions are included for human reference only and are not intended to be used by the operations developed in Chapter 4.

Although *Description* should not be needed to clarify a name where a CAP_{IC} Name is not easily human-understandable, *Description* may be used to provide human-understandable information. This allows ICs with similar human-understandable names to be uniquely identifiable for computational purposes. For example, somewhat obscure *Names* (e.g., BldgATel65, 312A, 34237, etc.) are difficult for a user to follow; however, the Description may clarify such labels.

In the case of a mobile system, there is an even greater concern with the uniqueness of *Name*(s). As a user moves from one environment to the next, there is an increased risk of being exposed to one or more CAP_{IC} Name(s) that are the same but describe very different ICs. In this case, users should always be advised of the need to update the CAP_{O} and given the choice of replacing an existing CAP_{IC}. Here the Description becomes an invaluable tool to assist users in renaming CAP(s) and/or determining whether a given CAP should be replaced.

3.1.2 Linkages

Linkages are used to create chains of interrelated records. All CAP specifications include information on linkages to other CAP specifications to which they are directly connected (i.e., higher to their parents, to their peers/siblings, and lower to their children). Table 3.5 illustrates the basic structure of CAP Linkages that will be elaborated on for each specific type of CAP.

Table 3.5: Linkages

	Description	*Possible Values*
Linkages		
Higher-CAP	Linkages to immediate higher-level CAP(s).	See following sections.
Peer-CAP	Linkages to CAP(s) at the same level with which interactions occur.	See following sections.
Lower-CAP	Linkages to immediate lower-level CAP(s).	See following sections.

Since a CAP_{O} is the highest level CAP, it has no parent linkages. All other CAP(s) should contain one (or more) linkage(s) to the next higher level CAP to which they belong.

Linkages between a CAP and other CAP(s) at the same level should only be specified for other CAP(s) with which this CAP directly interacts. For example, if interactions move $CAP_{IR} \rightarrow CAP_{PF} \rightarrow CAP_{OT}$, then the CAP_{IR} and CAP_{OT} would only identify the CAP_{PF} as a sibling CAP, and the CAP_{PF} would identify both the CAP_{IR} and the CAP_{OT} as sibling CAPs.

All CAP(s), except CAP_{CF}(s), also have linkages to lower level CAP(s). This chapter will explain which linkages apply to a type of specification in the following sections.

3.1.3 Type-Specific Information

Type-Specific Information is currently required only for CFs. It is used to provide the unique details of the IC such as its supported modalities, languages, and capabilities. See Section 3.2.3 for more information.

3.2 CAP Specification Details

This section will discuss the details of the various types of CAP specifications.

3.2.1 CAP_O High Level CAP

A CAP_O contains a set of ICs. As illustrated in Table 3.6, a CAP_O is identified with a type, a unique name, and a description. CAP_O specifications also require a set of linkages to various CAP_{IC}(s). When comparing alternate configurations, multiple different CAP_O(s) can be specified using (or reusing) selected CAP_{IC}(s) from some common set of CAP_{IC}(s).

Type, in the case of a CAP_O specification, is set to CAP_O.

The CAP_O *Linkages* contains links to one or more CAP_{IC}(s) that is part of this CAP_O and may contain links to other CAP_O(s).

Table 3.6: CAP_O High Level Specification Format

	Description	*Possible Values*
Identification		
Type	The type of record.	`CAP`$_O$
Name	See Table 3.4.	
Description	See Table 3.4.	
Linkages		
Lower-CAP_{IC}	The CAP_{IC}(s) included in this CAP.	{*cap-ic-name*, *cap-ic-name*, ...}
Peer-CAP_O	Other related CAP_O(s). (Optional)	{*cap-o-name*, *cap-o-name*, ...}

3.2.2 Interacting Component CAP(s)

As noted in Chapter 2, systems, users, ATs, and environments can be considered ICs. Table 3.7 expands upon the general format of the CAP_{IC} specification that was introduced in Table 3.2.

The *Type* entry is one of CAP_{USE}, CAP_{SYS}, CAP_{AT}, or CAP_{ENV}. In the case of a CAP_{AT}, it should be noted that, by definition, an AT is a specialized form of a System. This suggests that one could either use an IC of type CAP_{SYS} or an IC of type CAP_{AT} to specify an AT. However,

Table 3.7: CAP_{IC} Specification General Format

	Description	*Possible Values*
Identification		
Type	The type {User, System, AT, Environment} of record for this CAP_{IC}.	one of {$\texttt{CAP}_{USE}$ $\texttt{CAP}_{SYS}$ $\texttt{CAP}_{AT}$ $\texttt{CAP}_{ENV}$}
Name	A descriptive name for the IC. This field is type-dependent.	any (must be unique within CAP)
Description	See Table 3.4.	
Linkages		
Higher-CAP_O	The CAP(s) to whom this IC belongs.	{*cap-o-name*, *cap-o-name*, ...}
Peer-CAP_{IC}	All connected CAP_{IC}(s).	{*cap-ic-name*, *cap-ic-name*, ...}
Lower-CAP_{IR}	All IR(s) this IC uses.	{*cap-ir-name*, *cap-ir-name*, ...}
Lower-CAP_{PF}	All PF(s) this IC uses.	{*cap-pf-name*, *cap-pf-name*, ...}
Lower-CAP_{OT}	All OT(s) this IC uses.	{*cap-ot-name*, *cap-ot-name*, ...}

as mentioned in Chapter 2, ATs are notable for their ability to transform content streams. Thus, CAP_{AT}(s) differ from CAP_{SYS}(s) in one respect – while it is entirely possible for a CAP_{SYS} record to omit any CAP_{PF} information or even for the system itself to not have a PF, it is required (by definition) for an AT to have a PF and therefore for a CAP_{AT} to record CAP_{PF} information. Further, the requirements of a CAP_{SYS} also apply to a CAP_{AT}, with the exception that the *Type* field contains CAP_{AT}.

The *Name* entry of a CAP_{IC} is contextually dependent upon the *Type* entry. The *Name* entry of a CAP_{USE} should identify the user using the real name of the user or group of users, a username, or a nickname. The *Name* entry of a CAP_{SYS} should identify the software or hardware being described and, as such, the name used can be a trade name (brand name), trademark, or other commonly used name to describe this hardware / software product.

A CAP_{ENV} describes the environment or space in which other IC(s) are located and within which they must interact. Even when an IC is never physically moved, its environment may change. In the case of a CAP_{ENV}, the *Name* field can contain either an identifier of the space (e.g., "Room 312"), a commonly known name for the space (e.g., "Blue room", "Conference room", "Oval Office", etc.), or a user-defined user-specific name (e.g., "home", "office", etc.) providing it both uniquely identifies the environment and uniquely occurs in the CAP_O.

Although *Description* should not be needed to clarify a name, where a CAP_{ENV} Name is not easily human-understandable, *Description* may be used to provide a human-understandable environment location. This allows environments with similar human-understandable names to be uniquely identifiable for computational purposes. For example, confusion may occur on a campus where several buildings might have one or several rooms marked "Conference Room", consequently, to ensure uniqueness, their CAP_{ENV} Name might be somewhat obscure (e.g., BldgAConf15, 312A, 34487, etc.) and difficult for a user to follow, however the Description may clarify such labels. Similarly, in a mobile system, a user might move from an office building with a conference room to a meeting at a hotel with a conference room where the same CAP_{ENV} *Name* describes both rooms. The Description might help the user to make very different environments more unique in a CAP.

Among the linkages that can be contained in a CAP_{IC} specification, *Higher-*$\textit{CAP}_O$ refers to the one or more CAP(s) that contain this IC. As discussed in Section 3.1.2, *Lower-*$\textit{CAP}_{IR}$, *Lower-*$\textit{CAP}_{PF}$, and *Lower-*$\textit{CAP}_{OT}$ refer to the CF, if any, used by this CAP_{IC}. *Peer-*$\textit{CAP}_{IC}$ refers to all directly connected CAP_{IC}(s).

3.2.3 Component Feature CAP(s)

An IC is also specified by the sets of all CFs — IRs, PFs, and OTs — used by the system, AT, user, or environment the CAP_{IC} specifies. Every CAP_{IC} contains one or more CAP_{CF}(s) — that is, one or more CAP_{IR} / CAP_{OT} and zero, one or more CAP_{PF}. This section discusses information common to all CAP_{IR}, CAP_{PF}, and CAP_{OT}, as well as Type-Specific Information unique to a CAP_{PF}.

Table 3.8 expands upon the structure of the CAP_{CF} specification that is illustrated in Table 3.3. The specification of every CF has three major parts: CAP_{CF} Identification, CAP_{CF} Linkages, and CAP_{CF}-Specific Information.

Table 3.8: CAP_{CF} Specification General Format

	Description	*Possible Values*
Identification		
Type	The record type.	one of { CAP_{IR} CAP_{PF} CAP_{OT} }
Name	An identifier of, or a commonly known name for, the CF.	any (must be unique within CAP)
Description	See Table 3.4.	
Linkages		
Higher-CAP_{IC}	The IC(s) to whom this CF belongs.	{*cap-ic-name*, *cap-ic-name*, ...}
Peer-CAP_{IR}	The IRs used by this CF.	{*cap-ir-name*, *cap-ir-name*, ...}
Peer-CAP_{PF}	The PFs used by this CF.	{*cap-pf-name*, *cap-pf-name*, ...}
Peer-CAP_{OT}	The OTs used by this CF.	{*cap-ot-name*, *cap-ot-name*, ...}
Communication capabilities		
Capacity	The maximum number of channels the CF can accept.	{1 any other specific integer N}
Parallel	The number of channels capable of contiguous communication.	{0 any other specific integer N}
Serial	The number of channels capable of successive communication.	{0 any other specific integer N}
Degree	The number of channels connected to the CF.	{0 any other specific integer N}
Intermittent	The number of channels the CF is currently using for intermittent communication.	{0 any other specific integer N}

Table 3.8: (continued)

Devoted	Indicates whether one or more channels the CF is currently using for intermittent communication is available for the use of other CFs. Yes means that all intermittently used channels of the CF are “tied up”. No means that one or more of the intermittently used channels are available for other CFs to borrow.	one of {`YES, NO`}
Type-Specific Information		
CF Type-Specific Information		
Modality (see Section 3.2.3.1)		

ModalityType	The modality type of this CF. With the exception of ALL, multiple modalities require separate CF(s).	one of
		{`ALL`
		`VISUAL`
		`AUDITORY`
		`TACILE`
		`OLFACTORY`}

MediaTypes	All the media types used in this modality by this CF.	{`ALL`} or {`UNKNOWN`} or one or more of {`TextWritten`, `TextSpoken`, `TextSigned`, `TextTactile`, `Picture`, `VisualModel`, `Movie`, `DynamicVisualModel`, `Gesture`, `Sound`, `Music`, `Texture`, `TactileGraphic`, `ForceFeedback`, `Temperature`, `Odor`}
Language (see Section 3.2.3.2)		
Language	A list of pairs of all three-character ISO 639-3 language code(s) and four character ISO 15924 script code(s) in <*LanguageCode*, *ScriptCode*> format that apply to this CF. Keyword “NONE” may be substituted for the language code. Keyword “NIL” may be substituted for the script code.	{<abc, DEFG>, <def, NIL>, <NONE, ABCD>, <NONE, NIL>, ...}
Capability Specific Information(see specifics in Sections 3.2.3.3 and 3.2.3.4)		
Capabilities (for IR/OT and PF)		

Table 3.8: (continued)

Capability name	Has the value of the name of the capability. When the name "other" is used as the first part of the name, the capability comes from other specifications and a short name or other reference to the specification can be added to the name.	any
Capability instance	To allow multiple capabilities with the same name, an instance number can be used. If no instance number is provided, the default is 0, otherwise one higher than the previous *Capability instance* value of the same *Capability name*.	0 (zero or higher number)
Capability values	Either: a) A list of three values of upper and lower values and units (if applicable) of the *Capability name* for this CF. Assigning the same value as both an upper and lower bound specifies a single value. b) Format dictated by Taxonomy of Capabilities. c) Format dictated by another specification.	{ <upper, lower,unit>, <upper, lower,unit>, ...} or {any}
Connectivity (PF only)		
PassThrough	Content passes through the PF unmodified.	one of {YES, NO}
Transformed	Content is output in modified form.	one of {YES, NO}
Transformations (PF only)		
MediaTransformation	The PF transforms the Media.	one of {YES, NO}
LangTransformation	The PF transforms the Language.	one of {YES, NO}

The *Type* entry, which states the CAP$_{CF}$ record type being specified, is one of CAP_{IR}, CAP_{PF}, or CAP_{OT}. The specified type of the CAP$_{CF}$ determines what further entries are valid for the CAP$_{CF}$-Specific Information used to describe this CF.

Among the linkages that can be contained in a CAP$_{CF}$ specification, *Higher-*CAP_{IC} refers to the one or more CAP$_{IC}$(s) that contain this CF. *Peer-*CAP_{IR}, *Peer-*CAP_{PF}, and *Peer-*CAP_{OT} refer to linkages among the CF(s), if any, used by this CAP$_{CF}$.

The communication capabilities via channels of the CF are described with the *Capacity*, *Parallel*, *Serial*, *Degree*, *Intermittent*, and *Devoted* attributes. All of these attributes, except *Capacity* and *Devoted*, can accept any positive real number less than or equal to *Capacity*. The value 'N' is reserved for use where the number of channels is innumerable. This special value is most applicable to a CAP$_{USE}$ because users have many capabilities that can multitask.

For the CAP, further research needs to be conducted into the use of any attribute requiring quantitative measurement. This should include the validity of the measures and methods for obtaining them. Such quantitative measures may be more difficult to use with users than with systems.

Capacity cannot accept the value zero. The value zero would imply that the CF cannot accept any channels. However, a CF would not be documented in a CAP, if it has no capacity to accept channels. Thus, for *Capacity*, zero is not a valid value. Conversely, *Degree* is normally expected to document the use of at least one channel and could default to a minimum value of '1'. However, as shown in several examples in Chapter 2 (e.g., Figure 2.17), it is possible for a CF documented in a CAP to not use any channels because one of either the user or the system cannot interact using the modality of the CF. Thus, the value zero is valid for *Degree*.

Devoted describes the extent the intermittent channels of the CF are exclusively for its use. If there are channels available that can be "borrowed" by other CFs then these can be made temporarily available for their use. The value "YES" means that all intermittently-used channels of the CF are not available for the use of others. The value "NO" means that one or more of the intermittently-used channels are available for other CFs to borrow. Depending on how CAPs are managed in the future, it is possible that once all the available channels are used, this CF property could be automatically updated to "YES". However, such functionality is beyond the scope of this Thesis.

Note the attributes *Degree*, *Intermittent*, and *Devoted* are only set when a channel is being used by the CF. This will be discussed further in Chapter 5.

While it may be difficult to measure users sufficiently to provide values for *Capacity*, *Parallel*, and *Serial*, it is easier to measure these factors for the system. Knowledge of these values for a system may uncover accessibility issues. If provided with appropriate accessible tools, the user might be able to self-evaluate appropriate values for user attributes.

The *Parallel* and *Serial* attributes can help users in specific ways. While typical interface design choices may not bother most users, there are some cases where, for a user with a disability, it may not be obvious what is happening. For example:

- A user with very limited neck motion who uses a system with two screens, may not be able to easily switch screens. Using the Parallel attribute may help this user manage the visual output of the system.

- A user who is hard of hearing and has difficulty discriminating sounds may not be able to tell which sound represents which source when they are presented too closely together. Using the Serial attribute may help this user manage the auditory output of the system.

All CAP_{CF}(s) contain a variety of CAP_{CF}-Specific Information about the modality and language of the content stream. Note that although the CAP model refers to the idea of Direction, an indication of whether a communication is an input to or output of an IC, this is not listed in the CAP_{CF}. This decision was made because whether content is input or output can be deduced from the CAP_{CF} type: a CAP_{IR} describes an input, a CAP_{OT} describes an output.

The capabilities of a CF within a given modality and media type and/or the range of capabilities possible by the design of a CF is described in terms of Capabilities in a CAP_{CF}. Each CAP_{CF} can have multiple sets of Capabilities .

The remainder of this subsection will discuss Modality and Language CF-Specific Information and how IR/OT/PF Capability Specific Information is encoded in the CAP.

3.2.3.1 Modality CF-Specific Information

The Modality CF-Specific Information in Table 3.8 beginning on page 69 indicates how the content of the communication is presented (i.e., *ModalityType*) and describes the type of media used (i.e., *MediaTypes*). The *ModalityType* is either one of visual, auditory, tactile, olfactory, or the keyword ALL.

The structure of Modality CF-Specific Information in Table 3.8 includes identifying a basic modality of a CF, and the media used in this modality of the CF. The *MediaTypes* explicitly match the *ModalityType*s (e.g., the auditory modality does not support written text, but does support spoken text, music, etc.). Each CF has a single modality. Unless the *ModalityType* ALL is specified, multiple modalities of an IC are represented by multiple CF(s) being treated as separate parts of an IC. If and only if the *ModalityType* ALL is specified can the *MediaTypes* keyword ALL (which crosses multiple modalities of an IC) be used. Where a PF transforms modalities, the input modality is indicated in the Modality CF-Specific Information and the output modality information is indicated as the result of the transformation.

As Chapter 2 suggests, "noise" in an environment (as defined in this Thesis) includes olfactory noise. The current set of modalities supported by the CAP (i.e., visual, auditory, and tactile) is based on modern personal computer (PC) interfaces. An environment that has a known bad smell, which causes some user discomfort, or an environment that has known perfumes (e.g., industrial cleaners), which may cause an allergic reaction for a specific user, may be of interest to IC(s) of type CAP_{ENV} and/or CAP_{USE}. This implies that, in addition to the PC interface modalities already supported by a CAP (i.e., visual, auditory, and tactile), there is a need to further add the olfactory modality for use with CAP_{ENV} and CAP_{USE} ICs only. Thus, "olfactory" has been included as a *ModalityType*, but is currently restricted to environments (to describe known noxious odours and potential allergens) and users (to describe known allergies). It is expected that, as methods of olfactory interaction are further developed or become more common, this restriction will be removed allowing inclusion of systems and ATs.

In the context of *ModalityType*, the keyword ALL means that the CF being described can support all relevant modalities (i.e., visual, auditory, tactile, olfactory). As will be seen in Chapter 5, this keyword is most relevant to user CAPs because it explicitly means that a specific user's capabilities support all modalities and does not require further description.

Table 3.9 details the various types of media and is the source of the entry labels used to describe *ModalityType* in Table 3.8. This table expands Table 2.1 by including an olfactory modality and adding the signed text medium. Research in applications of scent in human-computer interaction is ongoing; however, there is currently little accepted jargon for describing olfactory outputs. In addition, although scents can carry meaning, current olfactory applications (e.g., olfactory icons

Table 3.9: Subdivisions of "Media" Revisited

ModalityType	***MediaTypes***
Visual	*Text-based* written text signed text †
	Not Text-based pictures visual models dynamic visual models † movies † gestures †
Auditory	*Text-based* spoken text †
	Not Text-based sounds † music †
Tactile	*Text-based* Braille
	Not Text-based textures tactile graphics force feedback gestures temperature
Olfactory	

Note: Items marked † are "dynamic only".

and 'smicons' Kaye, 2001) are largely not text-based. For these reasons, Table 3.9 leaves the *Olfactory* row blank.

Modalities can include one or more media. The set of *MediaTypes* listed in Table 3.8 beginning on page 69 can be used to identify the various major types of media. The following list defines each type of media.

- TextWritten: A language-based medium of words presented in a written symbolic script either statically or dynamically, typically by a system on a screen or by a user on a keyboard.

- TextSpoken: A language-based medium of words spoken by the user or system (e.g., speech audio).
- TextSigned: A language-based medium of words presented visually in a signed language (e.g., signed video).
- TextTactile: A language-based medium of words presented in a tactile symbolic script either statically or dynamically, typically by a system on a tactile display or by a user on a specialized keyboard (e.g., Braille input or output).
- Picture: A static image presented by the system or loaded into the system by the user.
- VisualModel: An object that combines both data and functionality such that table or model data can be used to render a static image (e.g., graph).
- Movie: A dynamic image which is presented by the system or loaded into a system by a user which changes over time without the user being able to control any changes other than the timing of the presentation.
- DynamicVisualModel: An object that combines both data and functionality such that table or model data can be modified by the system or the user and can be used to render a dynamic image by the system (e.g., animation).
- Gesture: Movements of the user which may express an idea or meaning. Gestures may be either tactile or visual in sensation.
- Sound: Any media that can be heard by the system or user but does not necessarily have an associated meaning.
- Music: Sounds produced by the system or user arranged in time possessing a degree of melody, harmony, or rhythm.
- Texture: Variation of the intensity (feel) of a surface produced by the system or user such as its smoothness, coarseness, and regularity.
- TactileGraphic: A static image produced by the system or user presented in a manner appropriate for tactile exploration.
- ForceFeedback: A means of tactilely expressing by a system to a user an understanding of the three-dimensional structure, shape, orientation, movement, or viscosity of a virtual object through pressure.

- Temperature: A medium that affects the sense of its degree of hotness or coldness.
- Odor: A medium that affects the sense of smell.

The terms “visual models” and “dynamic visual models” used in Tables 3.8 and 3.9 describe objects such as certain types of graphs and animations that contain data accessible by AT. These objects combine both data and functionality such that table or model data can be used to render graphs or animations.

In the context of *MediaType*, the keyword ALL is allowed if and only if the *ModalityType* keyword ALL has been specified because use of this keyword crosses multiple modalities which is otherwise not permitted in a single CAP_{CF}.

In the context of *MediaType*, the keyword UNKNOWN is used only if the media type is not otherwise obvious. For example, a computer mouse uses the tactile modality. A user is required by the design of the mouse to apply force to move it and pressure to activate it. However, unless designed to support force feedback output, the mouse itself does not use any of the above types of media. If the mouse uses tactile feedback, then specifying this would require a CAP_{OT} to deal with this feedback capability as well as the CAP_{IR} to describe the input capability.

The content of a medium can be either text-based or non-text-based. Text-based content, such as written text, has potential for direct transformation into alternative formats and thus can be more readily rendered accessible to various users. Non-text-based content, such as a graph, may need to be paired with text-based descriptions describing both the content and its meaning. Depending on the technology available to transform spoken/signed text into written text, some text-based content (i.e., spoken, signed) may need to be paired with written text transcriptions.

If one only considers the visual and auditory modalities, there are, according to ISO 14915-3, three basic media types: *audio*, *still image*, and *moving image* (ISO, 2002a). A moving image medium is a “visual medium that is delivered at a rate that is judged by the human viewer to be a continuous image” (ISO, 2002a) such as a film or animation. A still image medium is a “visual medium that is not presented continuously, although frames may be shown in a sequence either controlled by the user or by the system with a time delay” (ISO, 2002a) such as a photograph, or slide show. This breakdown of visual media is useful to our understanding of the visual modality and informs Table 3.9 (e.g., pictures, models, movies, etc.).

ISO 14915-3 further breaks down the audio media type into three further media types – *realistic*, *non-realistic*, and *speech*. Audio is “realistic” if it perceived by the user to represent the natural world (e.g., a bird’s song) or not (e.g., music). Audio is “speech” if it is spoken (ISO, 2002a). Since this notion of realism seems dubious and, as Table 3.9 shows, speech audio is considered text-based, this further breakdown of the audio media type does not add to our understanding of the auditory modality and is ignored in the CAP specification.

Table 3.10 shows that there are more than just the three media types (audio, still image, and moving image) mentioned above. The tactile modality allows for two more media types: *heat* and *pressure*. The olfactory modality is considered better suited for slowly changing information rather than discrete events (Kaye, 2001) and thus offers an additional media type: *odour*.

Table 3.10: Examples of Media

ModalityType (see Table 3.8)	**MediaType** (see Table 3.8)	**Type** (as per ISO 14915-3 and discussion)	**Example**
Visual	TextWritten	Still image Moving image	static text marquee
Auditory	Audio	Music	music
Tactile	Temperature ForceFeedback	Heat Pressure	 Braille
Olfactory	Odor	Odour	smicons

Entries in the *MediaTypes* field of Table 3.8 are one of the *MediaTypes* noted in Table 3.9. As the examples listed in Table 3.10 suggest, the same medium in the same modality (e.g., written text) can be output in multiple media types (e.g., still image, moving image) to create different effects (e.g., static page, marquee).

3.2.3.2 Language CF-Specific Information

Identification of the languages the IC can use is optional. Entries in the Language CF-Specific Information fields of Table 3.8 outline the languages and scripts supported. The language and script used may be derived from the language and script of the specification (see Appendix C, Section C.1 for more information).

The Language entry consists of a list of pairs of <*LanguageCode*, *ScriptCode*> that uniquely identify the supported languages/scripts used. This scheme ensures that, where multiple languages and scripts are indicated, the "correct" script is always associated with the "correct" language. That is, the script(s) used to represent a certain language is paired only with that language. This approach means that one cannot make the mistake of thinking a language (e.g., English) normally represented using one script (e.g., Latin), or its derivatives, is represented in another unlikely script (e.g., Cyrillic, Sanskrit, etc.). Therefore, this approach allows:

a) a user to state exactly which language/script pairs are preferred and/or can be used, and

b) content to be explicitly described (e.g., the message contains Ukrainian written using a Cyrillic script and English written using a Latin script).

With this approach, an example entry for English written text would be <eng, Latn>. However, some languages may require multiple entries for the same medium (e.g., Japanese written text could require four pairs, one each for Hiragana, Katakana, Kanji, and Latin). The remainder of this subsection discusses the specification of language and script codes.

Language Codes

A *LanguageCode* identifies a supported language. Such codes are characterized by the use of three-character language codes.

A *LanguageCode* entry in Table 3.8 may use the keyword NONE. The keyword NONE was chosen as a four-character keyword to avoid confusion with current and/or future three-character language codes. The keyword NONE is allowed for ICs with no language dependency of any kind (e.g., Environments can be language independent).

Where the *ModalityType* keyword ALL is used, it is not expected that the language specified in *LanguageCode* support or use all media types, only those media types the language is known to support. For example, signed languages use the visual and tactile modalities but do not use the auditory or olfactory modalities.

For two reasons, there is no *LanguageCode* keyword ALL. First, such a three-character keyword could be confused with an actual three-character language code. Second, in theory, the keyword ALL would require a system to support localization into every spoken, written, and signed language in the world or a user to be able to use and comprehend every spoken, written, and signed language in the world. For this reason, as a keyword, ALL makes no sense unless the described IC is truly culturally universal. However, being "culturally universal" would suggest no language dependency of any kind, a state, as mentioned above, for which the keyword NONE is used.

One challenge is deciding what three-character codeset to use and how to denote it. ISO and SIL International provide two possible choices.

ISO 639-2:1998 describes a three-character codeset to denote approximately 400 individual languages (International Organization for Standardization [ISO], 2002b). ISO 639-2 has several weaknesses:

- It has been noted for its limitations both in terms of the number of languages described (Simons, 2000) and for omitting whole classes of languages long recognized by linguists (Irish National Body, Deaf Action Committee for SignWriting, 1999).
- In its effort to be backwards compatible to the MARC bibliographic system (Byrum, 1999), ISO 639-2 allows 22 languages to have two alternative codes: bibliographic and terminology (e.g., French can be represented as either *fre* or *fra*).
- It is biased towards languages that have writing systems. A language does not appear in ISO 639-2 unless it has a known writing convention and associated body of literature (see sections 4.1.1, 4.1.3, and A.2.1 of ISO, 2002b).

- Since dialects of a language are usually represented in ISO 639-2 by the same language code as that used for the language, cultural sensitivities for mutually intelligible dialects are not well addressed (Constable & Simons, 2002).

- There is no distinction of languages based on choice of script (e.g., Serbian [*scc/srp*] is written in both the Cyrillic alphabet and Latin alphabet, Japanese [*jpn*] may be written using any or all of four different writing systems).

Current work to expand the existing ISO 639 standard from two to six parts is expected to resolve several of these issues (Dalby & Gillam, 2004).

Table 3.11 shows a dozen example ISO 639-2:1998 three-character codes. For languages with alternative codes, ISO 639-2/B (bibliographic) is given first followed by a slash ("/") and the ISO 639-2/T (terminology) code. Traditionally, ISO standards and specifications have shown a preference for the use of the terminology code.

Table 3.11: Example ISO 639-2 Language Codes

Language Name (English)	**ISO 639-2 code**	**Language Name (English)**	**ISO 639-2 code**
English	eng	French	fre/fra
Japanese	jpn	Korean	kor
Swedish	swe	Danish	dan
German	ger/deu	Klingon	tlh
Russian	rus	Ukrainian	ukr
Sign languages	sgn	Italian	ita

SIL International has released for public use a database of three-letter codes used in their *Ethnologue* (SIL International, 2006). This database is based on the reported observations of linguists and others around the world. Thus, the set of codes developed by SIL International offers a unique identifier for each of more than 7,000 individual languages. It includes all languages regardless of presence of a writing system or linguistic modality (written, spoken, signed). It also includes several dialects, pidgins and "grafted" languages (e.g., Signed English mostly grafts American Sign Language signs onto English grammar). Because of its relative completeness, the SIL system is already in use by several open standards (Simons, 2000). Recognizing this expertise as well as the weaknesses of the existing ISO 639-2 mentioned above, the new ISO 639-3:2007 is largely based on relevant parts of the SIL Ethnologue providing a list of six to seven thousand three-character language codes (Hjulstad, 2003; International Organization for Standardization [ISO], 2007a).

Thus, the codeset specified by ISO 639-3:2007 will be used to specify *LanguageCode*s

Script Codes

As mentioned above, some languages (e.g., Serbian, Japanese) can be written in multiple scripts. In addition, different users use different writing systems, and some users are comfortable using more than one writing system.

A *ScriptCode* identifies a supported script. Such codes are characterized by the use of four-character script codes.

A *ScriptCode* entry in Table 3.8 may also use the keyword NIL. The keyword NIL was chosen as a three-character keyword to avoid confusion with current and/or future four-character script codes. The keyword NIL is allowed for media with no script dependency of any kind (e.g., audiotext, pictures, etc.).

As with *LanguageCode*, where the *ModalityType* keyword ALL is used, it is not expected that the script specified in *ScriptCode* support or use all media types, only those media types the script is known to support. For example, many scripts use the visual modality but there is currently no known writing system that uses the olfactory modality.

ISO 15924:2004 provides a list of codes for the representation of names of scripts (International Organization for Standardization [ISO], 2004a). Each script can be represented as either a three-digit number or a four-character code. For example, the Latin script used in writing English can be represented as either *Latn* or *215*. This set of codes allows, in the case of written text, the ability to denote not only what language the text is written in (e.g., using systems like ISO 639-3), but also what script was used. This also allows users to specify scripts that they are able to read. For example, since Serbian [*scc/srp*] is written in both the Cyrillic alphabet and Latin alphabet, an IC that presents text content written in Serbian can specify whether the Cyrillic alphabet or Latin alphabet is used. Similarly, a user who reads Serbian can denote either the ability to read both scripts or the preference of one script over another. Table 3.12 shows ten example ISO 15924 codes. For simplicity of use and consistency with *LanguageCode*, *ScriptCode* entries use the four-character script codes of ISO 15924 rather than the equivalent three-digit codes.

Table 3.12: Example ISO 15924 Script Codes

English Name	**ISO 15924**		**English Name**	**ISO 15924**	
	number	**code**		**number**	**code**
Latin	215	Latn	Hangul	286	Hang
Latin (Gælic variant)	216	Latg	Blissymbols	550	Blis
Latin (Fraktur variant)	217	Latf	Braille	570	Brai
Cyrillic	220	Cyrl	Hiragana *	410	Hira
Cyrillic (Old Church Slavonic)	221	Cyrs	Katakana *	411	Kata

* As of 2006-06-21, the new code *Jpan* and number *413* have been added to provide an alias for Han + Hiragana + Katakana.

ISO 15924:2004 has a few weaknesses. For example, to meet the needs of users from different cultures, there are multiple codesets for Braille (e.g., US Braille, Japanese Braille, etc. see (Winter, n.d.)) as well as multiple formats (i.e., 6-dot Braille, 8-dot Braille); however, unlike Cyrillic and other writing systems that have similar requirements, these Braille codesets are not separately encoded. In addition, the list provided in ISO 15924 is incomplete and several known writing systems should be included (see Ager (1998) for a list of other known writing systems). Although Braille is listed, other tactile writing systems (e.g., the Moon alphabet) are not. In addition, ISO 15924 does not include methods for writing sign languages (e.g., SignWriting). At the very least, ISO 15924 should include the writing systems of every language listed in ISO 639-2 (e.g., Klingon).

In summary, *LanguageCode* is specified either with an appropriate ISO 639-3 three-character alpha code or NONE, and ScriptCode is specified either with an appropriate ISO 15924 four-character alpha code or NIL.

If desired, the CAP_{IC} can be expanded to specify additional information such as:

- whether only a specific dialect, grapholect, or culturally-specific lexicon of a language is used by a specific IC, or
- the precise fluency of a specific IC in a particular language or language skill (e.g., literacy, oral comprehension, etc.).

See Section 3.2.4 for more information.

3.2.3.3 Syntax for Adding Capability-Specific Information to the CAP

As mentioned in earlier chapters, the CAP is specific to capabilities that can be mapped between users and systems. On the one hand, this avoids a medical model focus on the user and their disabilities while on the other hand also avoiding a "design specific" requirements description of system-side technology. It allows the CAP to work with both current and future technologies.

Sources of Capability Specific Information

Capability specific information can come from several sources. Currently, these sources include:

- This Thesis.
- Current and future versions of the ISO JTC1 Special Working Group on Accessibility *User Needs Summary* (UNS) (ISO JTC 1 SWG Accessibility, 2006).
- Other specifications such as ISO/IEC 24751, a eight-part ISO standard based on Instructional Management Systems (IMS) Global Learning Consortium, Inc.'s Accessibility for Learner Information Package (LIP) specification (International Organization for Standardization [ISO] and International Electrotechnical Commission [IEC], 2006e, 2006c, 2006d).

- Other future sources.

The current discussion on capabilities is based on initial Chapter 5 work and additional analysis of the original Chapter 3 work.

The *User Needs Summary* (UNS), is a document prepared by the *ISO/IEC JTC1 Special Working Group on Accessibility* (ISO JTC 1 SWG Accessibility, 2006)[1]. Its purpose is to identify user needs and not to prioritize them or make any judgement regarding requirements for standards developers or product manufacturers. It provides a starting point for the analysis of existing standards to determine if gaps may exist. To meet the needs of this future "gap analysis", the UNS is focused on the need to identify capabilities (i.e., needs) rather than technology solutions (i.e., designs). Its scope includes:

- Computer hardware and software
- Web accessibility
- Communication technology
- Consumer electronics
- Electronic office equipment
- Public access terminals
- Broadcasting
- Electronic medical equipment
- Biometric systems
- Industrial equipment
- Transport information technology
- Educational technology
- Electronic document formats

Two example user needs from UNS are: "Some need … Visual information also available in auditory mode"; and "Some need … Information within viewable range of those of short stature or seated in wheelchairs" (ISO JTC 1 SWG Accessibility, 2006).

[1]While this Thesis was being examined, the status of the UNS changed. It is now being considered for publication as a three-part ISO/IEC Technical Report (International Organization for Standardization [ISO] and International Electrotechnical Commission [IEC], 2007b, 2007c, 2007d).

The UNS is suitable to capabilities because it describes, in general terms, a variety of user-specific needs. This document is not so much a source of codings but of potential capabilities that can be added to a CAP specification.

Another possible source is ISO/IEC 24751, a new eight-part standard under development by ISO JTC1/SC36, *Information Technology for Learning, Education and Training (ITLET)*, which is responsible for developing international standards for Learning Technology. Working Group Seven has the responsibility of developing standards addressing accessibility issues and is currently in the final stages of publishing ISO/IEC FDIS 24751-1, ISO/IEC FCD 24751-2, and ISO/IEC FCD 24751-3 to address individualized adaptability and accessibility in e-learning, education and training. These standards are based on the IMS ACCLIP (ACCessibility and Learner Information Profile) and ACCMD (Accessibility Metadata) Specifications.

The ACCLIP Specification provides a way for a learner's accessibility preferences with respect to display, control and content, to be recorded in a standardized manner. The context of this specification is described in terms of educational systems, in particular online systems such as WebCT (IMS Global Learning Consortium, Inc., 2003b).

The ACCMD Specification defines the meta-data that can be used to describe a learning resource's accessibility and its ability to match a learner's preferences, and works in conjunction with the ACCLIP Specification providing guidance on how to make them work together. By implementing both the ACCLIP Specification and the ACCMD Specification, e-learning systems should be able to determine what resources are accessible to which learners (IMS Global Learning Consortium, Inc., 2003a).

ISO/IEC 24751 defines a schema to specify accessibility preferences and learner accommodations. This specification is intended to help identify resources that match a user's stated preferences or needs. The needs and preferences addressed include:

- alternative presentations of resources,
- alternative methods of controlling resources,
- alternative equivalents to the resources themselves, and
- other enhancements or supports required by the user.

ISO/IEC 24751 consists at the moment of three parts:

1. Part 1 provides "a common framework to describe and specify learner needs and preferences on the one hand and the corresponding description of the digital learning resources on the other hand[,] so that individual learner preferences and needs can be matched with the appropriate user interface tools and digital learning resources" (ISO & IEC, 2006e).

2. Part 2 is about the Learners Accessibility needs and preferences, including:

- how digital resources are to be displayed and structured,
- how digital resources are to be controlled and operated, and
- what supplementary or alternative digital resources are to be supplied.

(ISO & IEC, 2006c)

3. Part 3 is about the description of the characteristics of the resource that affect how it can be perceived, understood or interacted with by a user, including:

   - what sensory modalities are used in the resource,
   - the ways in which the resource is adaptable (i.e., whether text can be transformed automatically),
   - the methods of input the resource accepts, and
   - the available alternatives (ISO & IEC, 2006d).

The work of SC36 and IMS means that providers of information should provide an alternative representation of the resource, to facilitate the different needs of the user[2].

This standard is suitable to capabilities because it specifically focuses on user capabilities such as left-handedness, need for screen reading, etc. While, ISO/IEC 24751 can be a source of specific codings of capabilities for use in a CAP_{CF}, the standard is very user-side oriented, thus codings may not have system-side equivalents as are favoured by the CAP. The use of ISO/IEC 24751 encodings within the CAP will require a considerably greater amount of information to be specified about various systems and their components than is needed within a basic CAP.

Any future resources of similar information can also be a source of capability information. This includes future changes that may occur to the above specifications from time to time as well as new specifications and resources that may appear.

Capabilities

Table 3.8 shows three specific fields (*Capability name*, *Capability instance*, and *Capability values*) for documenting capabilities within Capability-Specific Information shared by CAP_{IR}s, CAP_{OT}s, and CAP_{PF}s.

CAPs document the capabilities of an IC rather than constraints. Interactions are constrained by their need to involve matching capabilities. Thus, the only constraints of interest are the handicaps that occur when interactions are not able to occur. Capabilities documented in a CAP_{PF} can describe needed limitations to the range of capabilities of the CF. This allows content to meet the needs of users and other ICs who cannot perceive and/or use certain modalities.

[2]This is a potentially huge task for content providers. For this reason, the CAP is envisioned to have both manufacturer and community participation.

Capability name: Each capability has a *Capability name* value. The name "other" is used as the first part of the name for capabilities and needs identified in other specifications. A short name or other reference to the external specification can be added to the name (e.g. "other-ISO/IEC24751"). Capability names can also be derived from the Taxonomy of Capabilities (to be discussed).

Capability instance: To enable multiple capabilities with the same name, an instance number can be used. If no instance number is provided, the default value is zero (0); otherwise the value is one higher than the previous *Capability instance* value of the same *Capability name*. Thus, each new instance of a capability with the same Capability name will have a Capability instance incremented by one (1).

Capability values: The field *Capability values* has more than one valid entry format which is dependent on one of three cases:

a) The default format is a list of triples of upper and lower values and, where applicable, units of measure of the capability identified in *Capability name* for this $\text{CAP}_{IR}/\text{CAP}_{OT}$. Assigning the same value as both an upper and lower bound specifies a single value. Any well-recognized *Système International* (SI) unit of measurement (e.g., metre, gram, second, ampere, etc.) or any units outside the SI that are accepted for use with the SI (e.g., minute, bel, astronomical unit, tonne, etc.) or any well recognized computing unit (e.g., pixel) can be used as per ISO 80000 Parts 1-14 (International Organization for Standardization [ISO], 2006d, 2006e, 2006a, 2006b, 2007b, 2007j, 2007f, 2007c, 2007g, 2006c, 2007d, 2007e, 2007h, 2007i). Where no unit of measure applies, *unit* is set to NIL.

b) Where the *Capability name* comes from the Taxonomy of Capabilities, the format of *Capability values* is dictated by the taxonomy. This case will be described in more detail below.

c) Where *Capability name* includes the name "other" as the first part of the name for the capability, the format of *Capability values* is dictated by another specification also indicated in the *Capability name*. This case will be described in more detail below.

It is possible to have multiple sets of {*Capability name*, *Capability instance*, and *Capability values*} entries in individual CAP_{IR}s, CAP_{OT}s, or CAP_{PF}s. This allows the documentation of several not necessarily related capabilities for any one IC at once. This syntax provides the CAP an additive evolving approach to document capabilities.

The *Taxonomy of Capabilities* is a future data glossary, for specific capabilities not otherwise detailed in an ISO specification, such as ISO/IEC 24751, or other standard. The CAP tends to be top-down in its descriptive approach. This allows describing an IC as having all of a capability, part of it, or nothing. The Taxonomy of Capabilities glossary will allow CAPs to describe "sub-capabilities". For example:

- A user who has only monaural hearing still has the ability to receive auditory modality input but cannot perceive stereo-encoded information (e.g., source of sound). This capability could be denoted with the *Capability name* “Monaural” and a *Capability value* specifying which ear can hear. The Capability name “Monaural” might be defined by this future Taxonomy of Capabilities as having specific meaning for a system (e.g., do not output stereo sound) or leads to specific system actions (e.g., not encode information through stereo sound).

- A user who only has limited head and neck movement still has the ability to receive or transmit tactile information. This future taxonomy could define specific parts of the head and neck that, if used in a Capability name, has specific meaning for the system (e.g., use a specific set of pressure settings for any tactile output) or leads to specific system actions (e.g., determine if an onscreen keyboard is needed).

This proposed taxonomy will have several other similar examples. More information and examples will be discussed in Chapter 5.

PF Specific Information

PFs add to the syntax shared by $\text{CAP}_{IR}/\text{CAP}_{OT}$ by expanding it to document how content travels through the PF and the types of transformations that occur. Capability Specific Information in a CAP_{CF} describes the transformations, if any, in a PF being made on content streams. PF transformations can occur in terms of changes to any of the CAP_{CF} Capability Specific Information of an IR or OT.

The *Connectivity* of a CAP_{PF} only describes the high-level flow of the input from the IR(s) involved to the output of the OT(s) involved. In particular, Connectivity documents whether the input content stream is modified as it flows through the processing described by the CAP_{PF}. There are two basic connections: *pass-through* which is set to “yes” if the input content stream passes through the PF unmodified, and *transformed* which is set to “yes” if the input content stream is output in some modified form. Connectivity can describe four basic transformations:

a) no change (A $\rightarrow$ A) [*pass-through only*];

b) modify the input (A $\rightarrow$ B) [*transform only*];

c) modify and pass-through the input (A $\rightarrow$ A, B) [*both pass-through and transform*];

d) and no output (A $\rightarrow$ NULL) [*none*].

By connecting a number of PF(s) together, multiple new outputs can be produced.

The *Transformations* further describe the details of what CF-Specific Information is being changed: *Media* and/or *Language* and/or *Capabilities*. If the PF transforms the media properties of the content stream, the *MediaTransformation* field is set to "yes". If the PF transforms the Language properties of the content stream, the *LangTransformation* field is set to "yes". The existence of entries describing Capabilities is understood as a transformation of the capabilities of the content stream by default. Media transformations are, as noted above, further parameterized by PF Capability Specific attributes.

3.2.3.4 Examples of Capability Specific Information

As noted above, CAP_{IR}s and CAP_{OT}s document the capabilities of an IC to successfully interact with other ICs. Capabilities can be used to document specific user capabilities the system needs to match or specific system capabilities that may help or hinder the user.

This subsection presents and discusses some select examples of IR/OT capabilities. More examples are provided in Chapter 5.

Examples from this Thesis

Two example capabilities discussed in this Thesis are frequency and intensity.

Audio, visual, and tactile media can be described according to the frequencies used for their input/output. Each CAP_{IR} or CAP_{OT} capable of using such media is also only capable of handling certain frequencies and, as such, can have multiple sets of frequency capabilities. Capabilities in a CAP_{IR} or CAP_{OT} can be used to describe the range of frequencies used by content within a given modality and media type and/or the range of frequencies possible by the design of an IC.

Frequencies are expressed in Hertz. For example, humans can usually:

- hear sounds between 64 and 23000 Hz (Fay, 1988),
- perceive visible light (i.e., colour) between 384 and 769 THz (note: 1 THz = 10^{12} Hz) (Rossotti, 1983), and
- feel frequencies of vibration between 10 and 600 Hz (Fourney & Carter, 2005).

However, disabilities can reduce the frequencies that an individual can recognize/use. Documenting frequency capabilities in a CAP_{IR} or CAP_{OT} can describe the range of frequencies possible. For example:

- a tactile display may be designed to only output within a certain range of frequency.
- a visual display may be designed to be able to output sixteen million colours.

Table 3.13 shows how frequency capabilities can be specified in a CAP_{IR} or CAP_{OT}. Frequency capabilities are input in pairs of upper- and lower-frequency bounds. This allows the specification of specific frequencies as well as discrete ranges of frequencies.

Table 3.13: Frequency as an Example IR/OT Capability-Specific Information

Field	*Entry*	*Notes*
Capability name	`Frequency`	Name of Frequency capability.
Capability instance	`0`	First instance of a frequency capability in this IR/OT.
Capability values	{ <upper, lower, Hz>, <upper, lower, Hz>, ...}	A list of pairs of upper and lower frequency bounds in Hertz. Assigning the same value as both an upper and lower bound specifies a single frequency.

Since it is possible to have multiple sets of {*Capability name*, *Capability instance*, and *Capability Values*} entries in an individual CAP$_{IR}$ or CAP$_{OT}$, then several not necessarily related capabilities for any one IC can be documented at once.

For example, in addition to describing frequency capabilities, intensity capabilities may also be relevant. Audio, visual, and tactile media can also be described according to the intensity level of input/output. Each CAP$_{IR}$ or CAP$_{OT}$ capable of using such media is also only capable of handling certain intensities and, as such, can have multiple sets of intensity capabilities. Capabilities in a CAP$_{IR}$ or CAP$_{OT}$ can be used to describe the range of intensities used by content within a given modality and media type and/or the range of frequencies and intensities possible by the design of an IC.

Intensity refers to the magnitude of force or energy used per unit of surface, charge, mass, or time (Fourney & Carter, 2005). For example:

- auditory intensity is analogous to the acoustic notion of volume — the greater the intensity, the "louder" the experience of the stimulus (Fourney & Carter, 2005) — and is typically measured in decibels.

- visual intensity is analogous to brightness — the greater the intensity, the "brighter" the experience of the light — and is typically measured in candela.

Documenting intensity capabilities in a CAP$_{IR}$ or CAP$_{OT}$ can describe the range of intensities possible by the limitations of the design of an IC. For example, a PC speaker may be designed to only output sounds at 60 to 70 dB or a tactile display may be designed to only output within a certain range of intensity.

Table 3.14 provides an example of how a set of capabilities can be specified to document both frequency and intensity capabilities in a CAP$_{IR}$ or CAP$_{OT}$. Frequency and intensity capabilities are input in pairs of upper and lower frequency/intensity bounds. This allows the specification of specific frequencies/intensities as well as discrete ranges of frequencies/intensities.

Thus, the field values suggested in Tables 3.13 and 3.14 can be used as a basis for inputting future frequency/intensity capabilities with actual measurements.

Table 3.14: Frequency and Intensity as Example IR/OT Capability-Specific Information

Field	*Entry*	*Notes*
Capability name	`Frequency`	Name of Frequency capability.
Capability instance	`1`	Second instance of frequency capability in this IR/OT.
Capability values	`{ <upper, lower, Hz>, <upper, lower, Hz>, ...}`	A list of pairs of upper and lower frequency bounds in Hertz. Assigning the same value as both an upper and lower bound specifies a single frequency.
Capability name	`Intensity`	Name of Intensity capability.
Capability instance	`0`	First instance of intensity capability in this IR/OT.
Capability values	`{ <upper, lower, dB>, <upper, lower, dB>, ...}`	A list of pairs of upper and lower intensity bounds in decibels. Assigning the same value as both an upper and lower bound specifies a single intensity.

One capability determined during the analysis of Chapter 5 of this Thesis is related to timeouts. ISO 9241-20 recommends, "If a task requires users to make responses ... within a limited time in order for that response to be valid, the time range should be adjustable by the user, including the option to turn off all timing requirements" (International Organization for Standardization [ISO], 2007k). Timeouts can create difficulties in access for many kinds of users. As a result, it is useful to enable systems to document any timeouts and users to describe their own timeouts. Note that since ISO 9241-20 does not have its own specification, there is no need to use the keyword "other" when specifying this capability as shown in Table 3.15.

Table 3.15: Time-Out as an Example IR/OT Capability-Specific Information

Field	*Entry*	*Notes*
Capability name	`Timeout`	Name of Timeout capability.
Capability instance	`0`	First instance of timeout capability in this IR/OT.
Capability values	`{ <upper, lower, s>, <upper, lower, s>, ...}`	A list of pairs of upper and lower timeout bounds in seconds. Assigning the same value as both an upper and lower bound specifies a single interval of timeout.

Example from the User Needs Summary

An example capability derived from those discussed in the UNS is force. Requirements such as "Some need ... Method to fully operate product that does not require much force", and "Some need ... Method to fully operate product that does not require much sustained force" (ISO JTC

1 SWG Accessibility, 2006) suggest that force is a issue that can be documented as both a user and system capability. For users, some need to be able to push, pull, or twist system devices such as a mouse with as little force as possible. For systems, some have devices such as force feedback joysticks which resist the user so as to communicate tactile/haptic information.

Capabilities in a CAP_{IR} or CAP_{OT} can be used to describe the degree of force required for input with an IC or the levels of force output possible by the design of an IC. Force is expressed in newtons (N). For example, moving a computer mouse requires approximately 0.8N (Akamatsu & MacKenzie, 1996).

Documenting force capabilities in a CAP_{IR} or CAP_{OT} can describe the range of force possible by the limitations of the design of an IC. For example:

- a computer mouse may require a minimum amount of force to be moved over a flat surface.
- a device may require a certain amount of force (i.e., against it's weight) to carry it.

Table 3.16 provides an example of how a set of capabilities can be specified to document force in a CAP_{IR} or CAP_{OT}. Force capabilities are input in pairs of upper and lower bounds. This allows the specification of a specific force as well as discrete ranges of force.

Table 3.16: Force as an Example IR/OT Capability-Specific Information

Field	*Entry*	*Notes*
Capability name	`Force`	Name of Force capability.
Capability instance	`0`	First instance of force capability in this IR/OT.
Capability values	`{ <upper, lower, N>, <upper, lower, N>, ...}`	A list of pairs of upper and lower bounds in newtons. Assigning the same value as both an upper and lower bound specifies a single degree of force.

Thus, the field values suggested in Table 3.16 can be used as a basis for inputting future force capabilities with actual measurements.

Example from Other Specifications

As shown in Table 3.8, where *Capability name* includes the name "other" as the first part of the name for the capability, the format of *Capability values* is dictated by another specification also indicated in the *Capability name*. In this manner, CAPs can include fields and/or data from other specifications.

For example, one capability derived from ISO/IEC 24751 is a need for specialized cursor properties such as a specific size, colour, or the presence of cursor tails. Unlike CAP CF Type-Specific Capabilities, which use labels that clearly show user and system capabilities, which are always considered and documented, this constraint is a user-side capability requiring an explicit system-side response. As such, cursor requirements are a capability that can be documented as a user capability in terms of colour, magnification, and other requirements or a system capability in terms of specific system settings. For users, some need to be able to see or track system cursors, such as a mouse pointer. For systems, some have software that directly support changes to the cursors.

Table 3.17 provides an example of how, instead of using the default format of the *Capability values* field described above, an alternative format, based on an external specification (i.e., ISO/IEC 24751), is used to document cursor capabilities in a CAP_{IR} or CAP_{OT}. The *Capability values* field is input solely through XML code. Note that, in the interests of space and readability, only a partial ISO/IEC 24751 specification is shown (see IMS Global Learning Consortium, Inc., 2003c).

Table 3.17: Cursor Requirements as an Example IR/OT Capability-Specific Information

Field	*Entry*	*Notes*
Capability name	`other-ISO/IEC24751-cursors`	Cursor capabilities based on ISO/IEC 24751.
Capability instance	`0`	First instance of cursor capability in this IR/OT.
Capability values	<?xml version = "1.0" encoding = "UTF-8"?> <accessForAll xmlns = "http://www.imsglobal.org/xsd/ accessforallv1p0" xmlns:xsi="http://www.w3.org/2001/ XMLSchema-instance" xsi:schemaLocation= "http://www.imsglobal.org/ xsd/AccessForAll_v1p0.xsd"> <context identifier="ID000000" lang="en"> <display> <screenEnhance> <screenEnhanceGeneric> <cursorSize usage="preferred" value="0.5"/> <cursorColor usage="preferred" value="ffffffff"/> <cursorTrails usage="preferred" value="0.5"/> </screenEnhanceGeneric> </screenEnhance> </display> </context> </accessForAll>	Any external specification valid encoding can be provided.

Examples of PF Specific Information

As noted above, CAP_{PF}s document transformations, in particular needed changes / limitations to the range of capabilities of the transformed CF. This allows transformed content to meet the needs of users and other ICs who cannot perceive and/or use certain modalities.

Limitations in human perception of frequency are experienced as colour-blindness, hearing loss, and tactile numbness.

For example, a CAP_{PF} for a visual display IC, whose CAP_{CF} allows sixteen million colours, might specify frequency limitations such that only 256 colours can be displayed. Furthermore, specific frequencies for red and green colours might be transformed into other frequency ranges.

For example, a user may need soft audio content made louder. Thus, a CAP_{PF} for an IC, whose CAP_{CF} uses auditory text in the 40 to 65 dB range, might specify intensity requirements that limit the volume of the audio output to a range between 60 and 75 dB by processing the existing audio content up or down in volume as appropriate. Furthermore, specific higher-frequency sounds might be transformed into lower-frequency sounds.

Table 3.18 shows how the specification of *Capability name*, *Capability instance*, and *Capability values* described above are followed to document frequency capabilities in a CAP_{PF}. Frequency capabilities are input in pairs of upper- and lower-frequency bounds. This allows the specification of specific frequencies as well as discrete ranges of frequencies.

Table 3.18: Frequency as an Example PF Capability-Specific Information

Field	*Entry*	*Notes*
Capability name	`Frequency`	Name of Frequency capability.
Capability instance	`0`	First instance of frequency capability in this PF.
Capability values	`{ <upper, lower, Hz>, <upper, lower, Hz>, ...}`	A list of pairs of upper and lower frequency bounds in Hertz. Assigning the same value as both an upper and lower bound specifies a single frequency.

Thus, the field values suggested in Table 3.18 can be used as a basis for inputting future frequency capabilities with actual measurements.

As described above, connectivity capabilities document whether the input content stream is modified as it flows through the processing described by the CAP_{PF}. For example, an AT that supports text-to-speech modifies written text input into spoken text output. If the AT is not used, then the connectivity is pass-through (i.e., written text output). If the written text is changed into spoken text and only the spoken text is output, then the connectivity is transform only. If the written text is changed into spoken text and both the written and spoken texts are output, then the connectivity is both pass-through and transform. Finally, if nothing is output, then the connectivity is none.

In addition, transformations can be documented to described whether media, language, or other capabilities are being modified. For example, the AT that supports text-to-speech and modifies written text input into spoken text output, discussed above, represents a media transformation (i.e., visual to auditory). In this case, its Capabilities would at least document both the ability to use text and the ability to output speech, thus providing the parameters of the media transformation.

3.2.4 Additional Properties

The CAP structure can be expanded to support additional and/or new functionality. Such properties expand Table 3.8 by adding additional CF Type-Specific fields. This is designed to allow future expansion of CAP_{IC}(s). Descriptions of any other properties added to a CAP_{IC} specification should be made available to the users of the specification by both internal documentation (i.e., using a Descriptions field) and external documentation (e.g., information available through a technical support website).

Examples of such expansions may include additional descriptive information for existing fields (e.g., adding dialect / grapholect information to Language CAP_{CF}-Specific Information) and the need to add or test new or experimental ICs.

Appendix D suggests how various CAP(s) could be expanded to include optional non-computable CF Type-Specific Information that could provide supplementary information beyond that required to determine accessibility issues.

3.3 Summary

This chapter described the information that is contained in each type of CAP. The above specification provides a syntax for describing the various levels of the CAP model. All CAP levels are described in terms of Identifying information such as *Name*, *Type*, and *Description*, as well as their relationships with other CAPs (i.e., *Linkages*). Figure 3.1 depicts the current structure of the CAP.

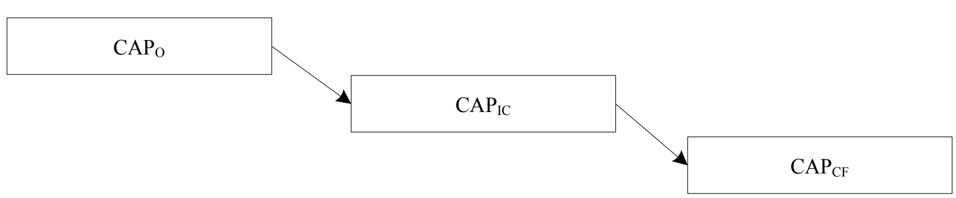


Figure 3.1: Visualization of the CAP structure.

- The highest level of the CAP, the CAP_O, is the root CAP of the "tree" of CAPs. It acts as a container of all lower level CAPs.

- The second level of the CAP, the CAP_{IC}, provides more detailed information about the users, systems, ATs, and environments documented within the CAP.

- The lowest level of the CAP, the CAP_{CF}, contains the majority of the information describing the IC including its supported Modalities, Languages, and other Capabilities that constrain the IC.

CF Type-Specific Information has been described in terms of existing International Standards (e.g., ISO 14915-3:2002, ISO 639-3:2007, ISO 15924:2004) and examples have been provided showing that the CAP can both describe capabilities identified by this Thesis, as well as capabilities identified by other sources (e.g., ISO/IEC 24751-1:2006) (ISO, 2002a, 2004a, 2007a; ISO & IEC, 2006e). If capabilities may apply, they must be documented. If the user documents a capability that the system cannot match, the system needs to ask the user what to do. The next chapter describes additional logic involved in specifying complex CAPs.

Chapter 4

Relationships within CAPs

The CAP is intended to help connect users and systems in a way that promotes an accessible computing experience. However, there are major differences between different users and systems that need to be overcome to make this accessible computing experience a reality.

CAPs of systems describe what the system needs (as input) and can do (as output). A system may be rigidly designed with little flexibility and may require its users to have capabilities that not everyone has. This same inflexibility can restrict the means available to output to users. For example, the system may require a user to provide keyboard input and perceive screen output. Systems may also be flexible and allow users a choice in the modalities used for interactions.

CAPs of users describe what the user needs and can do. Most users are flexible and have abilities that systems do not require (e.g., taste, smell). However, users may not have abilities that the system requires (e.g., manual dexterity). Flexibility can help users to access systems, if the system supports more than one input/output modality. For example, a user who cannot use a mouse might be able to use the keyboard if the system supports MouseKeys input. If not, the user must use some additional AT that supports alternative pointer input.

When the needs/capabilities of users and systems are incompatible, issues of accessibility arise. When used to describe what systems and users require and can do, CAPs can determine where issues of access might appear.

A system's inflexibility may produce demands that must be satisfied to use the system. For example, a system may only support screen output. Users may also have specific demands because of a disability. For example, a user with poor vision may need to use text-to-speech output.

Sometimes options are available for systems or users. This kind of flexibility is very applicable to users, but systems may have some flexibility too. For example, a user who can produce speech as well as use a keyboard could use a system that optionally supports speech-recognition input in addition to keyboard input. In addition, if a system can output text visually and/or auditorially, then a user who can hear as well as see a screen can choose visual output, audio output, or both visual and audio output.

Some capabilities of a system can interfere with the user so badly that it needs to be excluded from use. This kind of exclusion can apply to user control of some specific system IC. For example, a user might exclude all audio output because they are physically unable to hear.

Some capabilities of a system can be interfered with by the environment such that it affects the user so badly that the system capability needs to be excluded from use. For example, a user might exclude audio output because their environment is too noisy to hear the system.

This chapter focuses on individual CAP_{IC}s that profile a specific system, user, environment, or AT. Unary logical constructs (i.e., SHALL, MAY, NOT) can be used to qualify single CAP records as well as groups of records. Binary formal logic constructs (i.e., AND, OR, and XOR) provide a means to describe the operations involved in grouping/combining CAP specifications within a CAP_{IC}. This chapter will define and explore such unary and binary operations in the CAP.

4.1 Operations on CAPs

This section describes the various unary and binary operations available on CAP records.

4.1.1 Unary Operations

CAP unary (i.e., single argument) operators describe the properties of a CAP. Unary operators can occur on individual CAPs. In this sense, unary operators qualify CAPs.

4.1.1.1 Required (SHALL)

Suppose a system uses keyboard input. There is a need in a CAP of the system to qualify that keyboard input is required or necessary for the system to be accessible.

In the world of International Standards, a word often used to describe something that is necessary/required is “shall”. This term stands out from words like “should”, which suggest greater permissiveness, and “must”, which suggests intention, because “shall” implies inevitability and determination (i.e., something that *will* happen).

To express requirements that must be satisfied, either directly or indirectly, by a system, and/or AT, a CAP is prefixed by the SHALL qualifier.

The CAP unary operator SHALL expresses the notion that the CAP is **required**. This qualifier is mostly applicable to system and AT CAPs because it denotes what the system or AT needs. These needs are either input requirements (e.g., keyboard input is required) or output requirements (e.g., perceiving visual output is required).

The SHALL operator also applies to user CAPs but only in limited cases. Because the focus of the CAP is on accessibility, user CAPs must take care not to limit accessibility by making unnecessary requirements. SHALL can be used to explicitly express a user’s refusal to use other languages (e.g., based on exercising their legislated rights to their first language) even though other language skills are available.

The SHALL operator does not apply to environment CAPs. It should not be applied to environment CAPs because, although environments can preclude certain modalities, environments do not require certain forms of interaction.

Within the CAP structure, the SHALL operator is implied for systems and ATs.

4.1.1.2 Optional (MAY)

Suppose a user who can produce clear speech wants to use voice input instead of just a keyboard and mouse. The CAP_{USE} specifying the ability to use speech should state that speech is an optional method of interaction rather than a required one. There is a need to qualify this record as being optional, as far as the user is concerned, for accessibility to be achieved.

In the world of International Standards, a word often used to describe something that is permitted/allowed is "may". This term stands out from words like "should", which suggests much less permissiveness, and "can", which refers to the ability of the user, because "may" signifies permission.

The CAP unary operator MAY expresses the notion that the CAP is optional. It **may or may not occur**. This allows a user some choice in the use of specific ICs. This qualifier is mostly applicable to user CAPs because it denotes what the user is capable of. These capabilities are either system/AT input oriented (e.g., can use keyboard) or system/AT output oriented (e.g., can see video screen).

The MAY operator can be applied to system or AT CAPs. It should not be applied to system or AT CAPs which describe requirements on the user because the user needs to know what types of interactions will occur without any surprises. On the other hand, it can be used to describe parts of the system or AT that are not required of the user but available for use. For example, a mouse connected to a system which can be fully accessed via the keyboard alone would be described with the MAY operator since the mouse is not required to use the system. MAY does not apply to making choices, which is a binary or greater operation.

The MAY operator can be applied to environment CAPs because, although environments can preclude certain modalities (which can be documented by their omission), and environments do not require certain forms of interaction, users and systems may still need to interact via a specific modality if it is at all possible due to their own capabilities. The capabilities of a user and/or system may be so dependent on a specific modality, even if it is significantly impacted by the environment, that few alternatives exist.

Within the CAP structure, the MAY operator is implied for users.

4.1.1.3 Exclusion (NOT)

Suppose a user does not have the hand-eye coordination to easily use a pointing device. The user will want to exclude their use. Since CAPs describe what users need and can do, the CAP of the user can only specify the ability to use pointing devices, not the inability. However, if a system is aware that pointing devices are explicitly excluded, it might be able to present other options. There is a need in a CAP of the user to qualify that specific pointing devices must be explicitly excluded for the system to be accessible.

Environments also have the need to explicitly exclude methods of interaction for the system to be accessible within the context of the environment. Suppose an environment is noisy to the point that users need to wear hearing protection. The environment will need to exclude the use of any system/AT audio output. There is a need in a CAP of the environment to qualify that audio output must be explicitly excluded for the system to be accessible.

In the world of International Standards, a word often used to describe something that is not permitted/allowed is "not" or "shall not". This term expressly excludes permission to do something.

The CAP unary operator NOT expresses the notion that this CAP **shall not occur**. This qualifier is mostly applicable to users and environments because it denotes the environment's limitations and the user's capabilities. This operator expresses where a single CAP cannot be used and can allow a user or environment to force the system to *not* use specific ICs for better accessibility.

The NOT operator does not usually apply to system or AT CAPs because it is typically not needed. There are some exceptions. One exception is the case of when a system component is connected to the system but not available, as would normally be expected, due to some malfunction (e.g., corrupted drivers, not powered, etc.). The CAP_{IC} describing this component as part of the system would still be in the CAP_{O} because it is still part of the larger system. Using NOT allows the CAP_{O} to be updated to show the component is not available even though still connected/detected without deleting the original CAP_{IC} record.

Note that the NOT operator is a single operator and is not the combination of "SHALL" and "NOT". Use of combinations of unary operators (e.g., SHALL NOT) could imply that CAPs could be prefaced with two qualifiers (e.g., SHALL NOT CAP2, MAY NOT CAP3). This combination could allow nonsensical qualifiers (e.g., SHALL MAY CAP1). Thus, combinations of unary operators are not allowed.

The NOT operator should never be used for user CAPs. Care needs to be taken with the use of NOT. On the one hand, the use of the NOT operator could be seen as moving the CAP closer to the medical model approach because the CAP is more oriented to describing the capabilities of systems and the abilities of users not the lack thereof. As a result, it could be argued that simple omission of a capability should be sufficient. On the other hand, the inclusion of NOT makes sense logically because the structure of traditional logic includes NOT. Thus, NOT is provided to be used in limited situations where necessary, but not using it is preferred. The NOT operator must be specified, if needed.

Within the CAP structure, the NOT operator can be applied to environments. Since the CAP is oriented to accessibility and is focused on the interactions between systems and users, environment CAPs are really only needed where they might preclude potential interaction styles. From the standpoint of the model upon which this Thesis is based, environment CAPs can use NOT because we only want to discuss / describe the environment, when it interferes with the interactions between systems and users.

4.1.1.4 Other Candidate Unary Operations

In the analysis leading to the above CAP unary operators, other concepts were considered and found not necessary. What follows is a discussion of these operators and an explanation for their exclusion.

Not Applicable

Suppose a CAP_{IC} is included within a CAP that describes an IC that is not part of the system and/or not useful to the larger set of CAP_{IC}s.

For example, as mentioned in Chapter 2, as a user continues their day-to-day activities, their mental and physical status changes too. Thus, user capabilities applicable in the morning are no longer applicable in late afternoon (or vice versa). At this point several elements of the CAP_{USE} may no longer apply.

This problem could also occur if, for example, a user decides to copy a CAP_{O} applicable to one system over to another not entirely identical system without using appropriate CAP tools to process it first. Several CAP_{IC}s within such a CAP_{O} would not be applicable to the new system.

In either situation it may be desirable to not delete these CAP_{IC} records but keep them within the larger CAP_{O}.

The proposed CAP unary operator *Not Applicable* was intended to indicate explicitly that the CAP is not applicable. However, the notion of Not Applicable, in particular its implication for CAP exclusion, is very close to the meaning of NOT. For instance, applying the second example above would simply mean prefixing any new CAP_{IC} that does not match the actual system configuration with a NOT to explicitly exclude it from consideration. Thus, this situation can be covered by use of NOT.

Not Available

Suppose an IC, such as the system's speakers, is connected to the system but not available, as would normally be expected, due to some malfunction (e.g., corrupted drivers, not powered, etc.). The

CAP_{IC} describing the speakers as part of the system would still be in the larger CAP_O because the speakers are still part of the larger system. The CAP_O would need to be updated to show the speakers are not available, even though they are still connected/detected without deleting the original CAP_{IC} record.

The proposed CAP unary operator *Not Available* was intended to indicate explicitly that the IC is not available for use even though it is included within the CAP_O. However, the notion of Not Available may be close enough to the meaning of NOT that it can be covered by NOT instead. For instance, applying the above example would simply mean prefixing the speaker's CAP with a NOT to explicitly exclude it from consideration. For this reason, this case has been added as an exception to the description of NOT in Section 4.1.1.3.

No Qualifier

The proposed CAP unary operator *No Qualifier* (NULL) was intended to indicate explicitly that the CAP_{IC} has no qualifier (e.g., SHALL, MAY, NOT). Not having a qualifier may mean:

a) the meaning of {SHALL, MAY, NOT} does not apply easily to this CAP_{IC} and/or this situation, or

b) no qualifier has ever been assigned (i.e., a new CAP_{IC} has been added or an existing CAP_{IC} has simply never been qualified).

It is likely that a CAP_{IC} prefixed by a NULL unary operator would simply not be used anyway. Thus, prefixing the CAP_{IC} with a NOT to explicitly exclude it from consideration is likely sufficient.

In addition, CAP-supporting tools are expected to include business rules to ensure that specific assumptions (e.g., the need to always have a qualifier) are always met before a CAP_{IC} is placed within a larger CAP_O.

4.1.1.5 Summary

In conclusion, Table 4.1 summarizes these unary operations and their applicability to the different kinds of CAPs.

Table 4.1: Summary of Unary Operators

	System	AT	Environment	User
SHALL	✓	✓		✓
MAY			✓	✓
NOT	✓	✓	✓	✓

4.1.2 Binary Operations on CAPs that Users Interact With

CAP binary (i.e., two arguments) operators describe conjunctions of CAP records within a CAP_{IC}. Using such conjunctions allows simplifying the structure of the CAP, when describing all the requirements and capabilities of an IC. Such conjunctions provide a means to group specifications within a single user/system/AT/environment CAP_{IC} and document connections between these specifications. Documenting the grouping or connecting of CAP_{IC} specifications will fully profile a system, a user, an environment, or an AT.

4.1.2.1 Included (AND)

The CAP binary operator AND expresses the notion that both operands of the conjunction are included together for access. That is, AND joins requirements. This conjunction is implicitly expressed by CAP linkages within CAP_{IC}s and is the default meaning of such linkages for system, AT, and environment CAPs.

Although, simultaneous use is beyond the scope of this Thesis, it is interesting to note that the AND operator also implies that simultaneous use of each CF is possible, whether they are actually used simultaneously or not. For example, consider the linkage necessary between the audio and video portions of a film. If both CAP1 (the audio) and CAP2 (the video) exist, and they are both required by the system, then presumably the conjunction SHALL CAP1 AND SHALL CAP2 is expressed and both specifications within the system's CAP_{IC} must be used. Thus, care must be taken with AND since the possibility of simultaneous use of the CFs involved implies the potential for conflict.

4.1.2.2 Substitutable (OR)

The CAP binary operator OR expresses the notion that, for accessibility, either operand is required but both may occur. That is, OR combines options. This conjunction is implicitly expressed by CAP linkages within CAP_{IC}s and is the default meaning of such linkages for user CAPs. OR can also be used by CAPs for users, systems, and ATs. The OR operator is not appropriate for environment CAPs because environments cannot be substituted.

The OR operator implies that the operand CAPs complement each other – they are substitutable. When IC specifications are described using this conjunction, it means that the content of one may be swapped in favour of the other. While the content represented by each component feature in the CAP_{IC} may not be exactly the same, they may be equivalent. The OR conjunction implicitly forbids discordant content – the content represented by each operand cannot be so different that they are not swappable. Thus OR is frequently used in the CAP_{IC} to designate equivalent alternatives ensuring that a user who cannot access the content represented by one operand is still able to access equivalent content through another. This allows a user some choice to increase accessibility.

For example, the audio and caption portions of a film are considered equivalent. They are not the same because they use different media. They are equivalent because each expresses fully the content of the other within its respective medium. Users can choose to use either the captions or audio soundtrack, or both simultaneously, if desired. If both CAP1 (the audio) and CAP2 (the captions) exist, and at least one of them is required to be presented by the system, then the conjunction SHALL CAP1 OR SHALL CAP2 is expressed and one or both specifications within the system's CAP_{IC} can be used.

4.1.2.3 Mutually Exclusive (XOR)

The CAP binary operator XOR expresses the mutual exclusion of CAP specifications and ensures that conflicts cannot occur. Traditionally, XOR expresses the notion that one of the operands is required, but both *can never* occur together. Mutual exclusion is often necessary to increase access by reducing channel overload. This conjunction is expressed by CAP linkages within CAP_{IC}s. XOR is not a default for any such linkages. XOR can be used by CAPs for systems, ATs, and users.

In the case of ATs, only those ATs in transformation mode can use the XOR operator. Recall, in transformation mode, content is transformed into a new medium and the original content stream is silenced. ATs in transformation mode are the only ATs allowed to use XOR because this use closely fits the definition of transformation.

In the case of users, XOR allows the explicit description of capability limitations. For example, a user unable to perform two actions at the same time, such as a chorded keypress (e.g., press both CONTROL and P keys simultaneously), might group CAPs within a CAP_{IC} that describe keypress capabilities with an XOR to exclude their simultaneous use. As another example, a user may want to control how many different audio channels their hearing needs to attend to and thus, use an XOR to control the presentation of multiple audio channels to control cognitive overloaded.

Although simultaneous use of two CAPs is not always a requirement when considering whether CAPs are mutually exclusive, mutual exclusion is often necessary to ensure that potentially conflicting content is not transmitted across the same channel simultaneously. For example, the audio and descriptive audio portions of a film are typically presented on different (auditory) channels, but, since both provide audiotext, users can only listen to one at a time. If both CAP1 (the audio soundtrack) and CAP2 (the descriptive audio including soundtrack) exist, and both are required but only one of them is allowed to be presented by the system, then the relationship SHALL CAP1 XOR SHALL CAP2 is expressed and only one specification within the system's CAP_{IC} is used.

4.1.2.4 Summary

In summary, Table 4.2 summarizes these binary operations and their applicability to the different kinds of CAPs.

Table 4.2: Summary of Binary Operators

	System	**AT**	**Environment**	**User**
AND	✓	✓	✓	
OR	✓	✓		✓
XOR	✓	✓		✓

4.2 Evolving the CAP Structure

The above discussion suggests that at least three changes are needed in the CAP framework developed in the previous chapter.

The **first** necessary change is the need to provide a space to document unary operators in the CAP. Since:

a) unary operators qualify the CAP record itself and not its relationships to other CAPs, and

b) unary operators are needed for all CAPs,

the *Qualifier* field has been added to the Identification portion of the CAP to document the CAP's applicable unary operator.

The **second** necessary change is the need to provide a space to document binary operators in the CAP. This can be achieved by adding a TYPE field to each linkage. Recall that linkages to higher/peer/lower CAPs are by the name of the target CAP. Given that:

a) there are at least three possible meanings to a linkage (i.e., AND, OR, XOR),

b) these linkages have specific default meanings depending on the IC type (i.e., AND in the case of system/AT/environment CAP_{IC}s, OR in the case of user CAP_{IC}s),

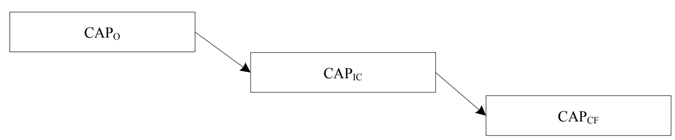
c) specific meanings cannot occur with specific IC types (i.e., AND does not apply to user CAP_{IC}s), and

d) XOR is never a default linkage type,

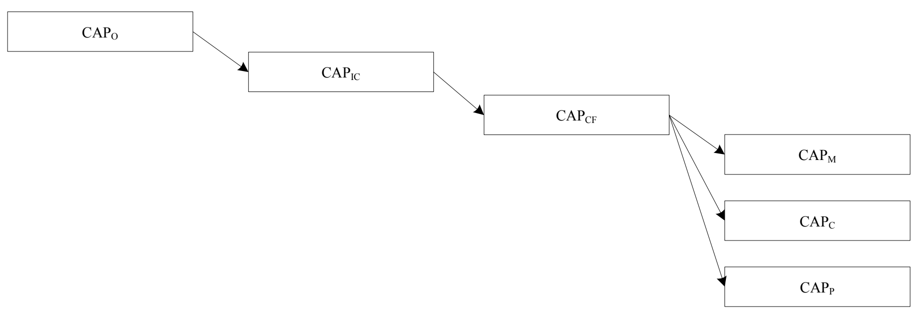
then the proposed TYPE field will need to at least address OR and XOR, since XOR must always be explicitly expressed and OR must be explicitly expressed for system, AT, and environment CAP_{IC}s specification linkages. Thus, Linkages will now be described as <*cap-name*, *linkage-type*> pairs.

The *cap-name* field will remain the name of the target CAP. The *linkage-type* field will describe the applicable binary operator this link implies (i.e., AND, OR, XOR), but if left blank will imply the default linkage type for the given IC type (i.e., AND in the case of CAP_{SYS}/CAP_{AT}, OR in the case of CAP_{USE}).

The **third** necessary change is to move the CAP framework from a three-level CAP_O / CAP_{IC} / CAP_{CF} structure (see Figure 4.1(a)) to a four-level structure where CF Type-Specific Information (i.e., Modality, Capability, and Processing) are handled within their own specific tables (i.e., CAP_M, CAP_C, and CAP_P, respectively). This is shown in Figure 4.1(b). This allows the use of linkages to describe the relationship of each concern with others within the same CAP. For example, it could allow linking two Modality records (i.e., a CAP_M) through XOR within the same CAP_{CF}. It also allows ignoring records that are not applicable leading to simplification and space saving. For example, it could allow not including a Processing record (i.e., a CAP_P), if no processing is being performed.



(a) The three-level CAP structure from Figure 3.1 on page 93.



(b) A four-level CAP structure.

Figure 4.1: The change of the CAP structure will move it from the existing three-level system described previously (a) to a new four-level structure (b).

Tables 4.3, 4.4, 4.5, and 4.6 show the new format for CAP_{CF} encodings.

Table 4.3: CAP_{CF} Specification General Format (based on Table 3.8 on page 69)

	Description	*Possible Values*
Identification		
Type	The record type.	one of {$\texttt{CAP}_{IR}$, $\texttt{CAP}_{PF}$, $\texttt{CAP}_{OT}$}
Name	An identifier of, or a commonly known name for, the CF.	`any` (must be unique within CAP)
Qualifier	A unary operator that qualifies this CAP as either being required, optional, or excluded. This field is required to always be completed; thus, the default setting at the time the CAP record is first created is MAY.	one of {`SHALL, MAY, NOT`}
Description	A narrative description to record preliminary information and / or optional comments further describing the object.	`any`
Linkages		
Higher-CAP_{IC}	The IC(s) to whom this CF belongs.	{<*cap-ic-name*, *linkage-type*>, <*cap-ic-name*, *linkage-type*>, ...}
Peer-CAP_{IR}	The IRs used by this CF.	{<*cap-ir-name*, *linkage-type*>, <*cap-ir-name*, *linkage-type*>, ...}
Peer-CAP_{PF}	The PFs used by this CF.	{<*cap-pf-name*, *linkage-type*>, <*cap-pf-name*, *linkage-type*>, ...}
Peer-CAP_{OT}	The OTs used by this CF.	{<*cap-ot-name*, *linkage-type*>, <*cap-ot-name*, *linkage-type*>, ...}
Lower-CAP_{M}	The modality specifications of this CF.	{<*cap-m-name*, *linkage-type*>, <*cap-m-name*, *linkage-type*>, ...}
Lower-CAP_{C}	The capabilities of this CF.	{<*cap-c-name*, *linkage-type*>, <*cap-c-name*, *linkage-type*>, ...}

Table 4.3: (continued)

Lower-CAP$_P$	The processing specifications of this CF.	{<*cap-p-name*, *linkage-type*>, <*cap-p-name*, *linkage-type*>, ...}
Communication capabilities		
Capacity	The maximum number of channels the CF can accept.	{`1` `any other specific integer` `N`}
Parallel	The number of channels capable of contiguous communication.	{`0` `any other specific integer` `N`}
Serial	The number of channels capable of successive communication.	{`0` `any other specific integer` `N`}
Degree	The number of channels connected to the CF.	{`0` `any other specific integer` `N`}
Intermittent	The number of channels the CF is currently using for intermittent communication.	{`0` `any other specific integer` `N`}
Devoted	Indicates whether one or more channels the CF is currently using for intermittent communication is available for the use of other CFs. Yes means that all intermittently used channels of the CF are "tied up". No means that one or more of the intermittently used channels are available for other CFs to borrow.	one of {`YES, NO`}

Table 4.4: CAP_{CF} Modality-Specific Information Identification

	Description	*Possible Values*
Identification		
Type	The record type.	$\texttt{CAP}_M$
Name	An identifier of, or a commonly known name for, the modality.	`any` (must be unique within CAP)
Qualifier	A unary operator that qualifies this CAP as either being required, optional, or excluded. This field is required to always be completed; thus, the default setting at the time the CAP record is first created is MAY.	one of {`SHALL, MAY, NOT`}
Description	See Table 4.3.	
Linkages		
Higher-CAP_{IR}	The IR(s) to whom this modality belongs.	{<*cap-ir-name*, *linkage-type*>, <*cap-ir-name*, *linkage-type*>, ...}
Higher-CAP_{OT}	The OT(s) to whom this modality belongs.	{<*cap-ot-name*, *linkage-type*>, <*cap-ot-name*, *linkage-type*>, ...}
Higher-CAP_{PF}	The PF(s) to whom this modality belongs.	{<*cap-pf-name*, *linkage-type*>, <*cap-pf-name*, *linkage-type*>, ...}
Peer-CAP_M	The other modalities used by this IR/OT/PF.	{<*cap-m-name*, *linkage-type*>, <*cap-m-name*, *linkage-type*>, ...}
Peer-CAP_C	The other capabilities used by this IR/OT/PF.	{<*cap-c-name*, *linkage-type*>, <*cap-c-name*, *linkage-type*>, ...}
Peer-CAP_P	The other processing specifications used by this IR/OT/PF.	{<*cap-p-name*, *linkage-type*>, <*cap-p-name*, *linkage-type*>, ...}
Modality-Specific Information		

Table 4.4: (continued)

ModalityType	The modality type of this CF. With the exception of ALL, multiple modalities require separate CF(s).	one of {`ALL` `VISUAL` `AUDITORY` `TACTILE` `OLFACTORY`}
MediaTypes	All the media types used in this modality by this CF.	{`ALL`} or {`UNKNOWN`} or one or more of {`TextWritten`, `TextSpoken`, `TextSigned`, `TextTactile`, `Picture`, `VisualModel`, `Movie`, `DynamicVisualModel`, `Gesture`, `Sound`, `Music`, `Texture`, `TactileGraphic`, `ForceFeedback`, `Temperature`, `Odor`}
Language	A list of pairs of all three character ISO 639-3 language code(s) and four character ISO 15924 script code(s) in <*LanguageCode*, *ScriptCode*> format that apply to this CF. Keyword "NONE" may be substituted for the language code. Keyword "NIL" may be substituted for the script code.	{<`abc, DEFG`>, <`def, NIL`>, <`NONE, ABCD`>, <`NONE, NIL`>, `...`}

Table 4.5: CAP_{CF} Capability-Specific Information Identification

	Description	*Possible Values*
Identification		
Type	The record type.	`CAP`$_C$
Name	An identifier of, or a commonly known name for, the capability Specification.	`any` (must be unique within CAP)
Qualifier	A unary operator that qualifies this CAP as either being required, optional, or excluded. This field is required to always be completed; thus, the default setting at the time the CAP record is first created is MAY.	one of {`SHALL, MAY, NOT`}
Description	See Table 4.3.	
Linkages		
Higher-CAP_{IR}	The IR(s) to whom this capability Specification belongs.	{<*cap-ir-name*, *linkage-type*>, <*cap-ir-name*, *linkage-type*>, ...}
Higher-CAP_{OT}	The OT(s) to whom this capability Specification belongs.	{<*cap-ot-name*, *linkage-type*>, <*cap-ot-name*, *linkage-type*>, ...}
Higher-CAP_{PF}	The PF(s) to whom this capability Specification belongs.	{<*cap-pf-name*, *linkage-type*>, <*cap-pf-name*, *linkage-type*>, ...}
Peer-CAP_M	The other modalities used by this IR/OT/PF.	{<*cap-m-name*, *linkage-type*>, <*cap-m-name*, *linkage-type*>, ...}
Peer-CAP_C	The other capabilities used by this IR/OT/PF.	{<*cap-c-name*, *linkage-type*>, <*cap-c-name*, *linkage-type*>, ...}
Peer-CAP_P	The other processing specifications used by this IR/OT/PF.	{<*cap-p-name*, *linkage-type*>, <*cap-p-name*, *linkage-type*>, ...}
IR/OT Capability-Specific Information		
Capabilities		

Table 4.5: (continued)

Capability name	The name of the capability. The name "other" is used as the first part of the name for capabilities coming from other specifications. A short name or other reference to the specification can be added to the name.	`any`
Capability instance	To allow multiple capabilities with the same name, an instance number can be used. If no instance number is provided, default is 0; otherwise, it is one higher than the previous *Capability instance* value of the same *Capability name*.	`0` (zero or higher number)
Capability values	Either: a) A list of three values of upper and lower values and units (if applicable) of the *Capability name* for this CF. Assigning the same value as both an upper and lower bound, specifies a single value. b) Format dictated by Taxonomy of Capabilities. c) Format dictated by another specification.	{<*upper, lower, unit*>, <*upper, lower, unit*>, ... } or `any`

Table 4.6: CAP_{CF} Processing-Specific Information Identification

	Description	*Possible Values*
Identification		
Type	The record type.	CAP_P
Name	An identifier of, or a commonly known name for, the Processing Specification.	`any` (must be unique within CAP)
Qualifier	A unary operator that qualifies this CAP as either being required, optional, or excluded. This field is required to always be completed; thus, the default setting at the time the CAP record is first created is MAY.	one of {`SHALL, MAY, NOT`}
Description	See Table 4.3.	
Linkages		
Higher-CAP_{IR}	The IR(s) to whom this Processing Specification belongs.	{<*cap-ir-name*, *linkage-type*>, <*cap-ir-name*, *linkage-type*>, ...}
Higher-CAP_{OT}	The OT(s) to whom this Processing Specification belongs.	{<*cap-ot-name*, *linkage-type*>, <*cap-ot-name*, *linkage-type*>, ...}
Higher-CAP_{PF}	The PF(s) to whom this Processing Specification belongs.	{<*cap-pf-name*, *linkage-type*>, <*cap-pf-name*, *linkage-type*>, ...}
Peer-CAP_M	The other modalities used by this IR/OT/PF.	{<*cap-m-name*, *linkage-type*>, <*cap-m-name*, *linkage-type*>, ...}
Peer-CAP_C	The other capabilities used by this IR/OT/PF.	{<*cap-c-name*, *linkage-type*>, <*cap-c-name*, *linkage-type*>, ...}
Peer-CAP_P	The other processing specifications used by this IR/OT/PF.	{<*cap-p-name*, *linkage-type*>, <*cap-p-name*, *linkage-type*>, ...}
Processing-Specific Information		
Capabilities		

Table 4.6: (continued)

Capability name	The name of the capability. The name "other" is used as the first part of the name for capabilities coming from other specifications. A short name or other reference to the specification can be added to the name.	`any`
Capability instance	To allow multiple capabilities with the same name, an instance number can be used. If no instance number is provided, default is 0; otherwise, it is one higher than the previous *Capability instance* value of the same *Capability name*.	`0` (zero or higher number)
Capability values	Either: a) A list of three values of upper and lower values and units (if applicable) of the *Capability name* for this PF. Assigning the same value as both an upper and lower bound, specifies a single value. b) Format dictated by Taxonomy of Capabilities. c) Format dictated by another specification.	{<*upper, lower, unit*>, <*upper, lower, unit*>, ...} or `any`
Connectivity		
PassThrough	Content passes through the PF unmodified.	one of {`YES, NO`}
Transformed	Content is output in modified form.	one of {`YES, NO`}
Transformations		
MediaTransformation	The PF transforms the Media.	one of {`YES, NO`}
LangTransformation	The PF transforms the Language.	one of {`YES, NO`}

With this change, *Type*, can now be any of CAP_O, CAP_{USE}, CAP_{SYS}, CAP_{AT}, CAP_{ENV}, CAP_{IR}, CAP_{OT}, CAP_{PF}, CAP_M, CAP_C, and CAP_P. Note that CAP_{IC} and CAP_{CF} are general templates used for multiple specific types of CAP(s) and are not appropriate to be used as a *Type*.

With these three changes to the CAP framework, the ideas in this chapter can be further explored using the example CAPs that will be developed in the next step of this Thesis.

Chapter 5

Example CAPs

The purpose of this chapter is to show that CAPs can be generated using the structure and specification developed in Chapter 3 and the binary and unary relationships developed in Chapter 4. As part of this goal, nine sample CAPs (four user, three system, and two environment) have been developed and are discussed below.

For the purposes of this chapter, the olfactory modality will be ignored since it is not currently widely used.

To start these CAPs, we need a CAP$_O$ to contain all the lower level CAPs. Recall in Chapter 4 that Linkages are described in terms of <*cap-name*, *linkage-type*> pairs and that, where the *linkage-type* is not specified, the default value for the record type is to be assumed. This is shown in Table 5.1.

Table 5.1: The CAP$_O$

CAP$_O$ Identification	
Name	Root
Type	CAP$_O$
Qualifier	MAY
Description	This is the root node for the various examples in this chapter.
CAP$_O$ Linkages	
Peer-CAP$_O$	
Lower-CAP$_{IC}$	{<Clarks_Cap, >, <Johanns_Cap, >, <Pams_Cap, >, <Taes_Cap, >, <StandardSystem, >, <AnotherStandardSystem, >, <SystemWithAT, >, <ConferenceHall, >, <Kitchen, >}

Table 5.1 documents the names of the example CAPs discussed in this chapter:

- Type is filled with the default entry for an overall CAP,
- Qualifier is "MAY" to indicate the optional use of this CAP$_O$,
- Peer-CAP$_O$ links to nothing because there are currently no peers to this record, and

- Lower-CAP_{IC} lists the names of all of the CAPs discussed in this chapter. Each of these CAPs will be introduced in the discussion below.

5.1 Users

This section describes four example users and the contents of their respective CAP_{USE}s.

There is no such thing as an "average" user. Similarly, there is no such thing as an "average" CAP_{USE}. As such, one cannot just start building example user CAPs without having a sense of a specific or generic user and their needs with respect to computer use. For this reason, there is a need to have specific rules necessary for the creation of user CAPs.

The following Production Rules are to be applied, when creating CAP_{USE}s:

1. If someone other than the User is creating the CAP_{USE}, fully interview the User to determine the capabilities they bring into an interaction with the system.

2. If someone other than the User is creating the CAP_{USE} and, during this interview process, the user identifies specific disabilities, then, to the fullest extent possible, change the statement of disability into a statement of ability.

3. Always assume full support for all modalities and media types, unless otherwise stated. Be prepared to document what does work.

4. CAP_{USE} names need to focus on modalities (auditory, visual, tactile, olfactory), not "body parts". For example, use "Visual" rather than "Sight".

5. CAP_{USE}s must use the data values described in Section 3.2.

6. If the ALL keyword is not appropriate (i.e., for ModalityType), consider all of the legal data values. It is possible that one or more of the data values cannot be used. While it may be that some data value will eventually not be included in a specific CAP_{USE}, it must still be considered. Specify each data value that can be used.

7. Capabilities can be described and documented in terms of the User's senses and other skills as per Chapter 3.

8. Because a preference is not an access issue, CAP_{USE}s cannot be used to document preferences. Care must be taken to ensure that a "requirement" is not a preference in disguise. Such preferences are not capabilities, but simply preferences, based on what a person is used to. Such "requirements" may be evidence of existing capabilities.

For the purpose of creating example CAP_{USE}s, two ways to describe a user's approach to a system are through use scenarios (i.e., user modeling) or through "personas".

Realistic scenarios are good design tools because they depict the work practices that one hopes to support. However, they are weak because they are not very engaging. In addition, scenarios are often difficult to reconstruct and hard to extend with confidence (Grudin & Pruitt, 2002).

Cooper (1999) argued that designing for any one external person is better than trying to design vaguely for everyone or specifically for oneself. As a result, he introduced, as an interaction design technique, the notion of "personas".

A persona is a profile of an archetypal user synthesized from a series of interviews with real people, and includes a name, photograph, description of skills, and a set of user goals that drive the design of the product or system (Cooper, 1999). By closely adhering to the goals of a specific persona, designers satisfy the needs of the many users who have goals similar to those of the persona (D'Souza & Lincoln, 2004).

Personas have several observed benefits (Cooper, 1999; Grudin & Pruitt, 2002; D'Souza & Lincoln, 2004) including:

- Creating a common vision of users across the design team and communicating user needs.
- Making the development process more user-driven.
- Avoiding designing for all users and using oneself (i.e., the software developer) as a reference.
- Defining the feature set for a particular scenario.

With personas, instead of discussing abstractions such as "average users" or working with difficult scenarios, designers are able to talk about concrete users with concrete goals, abilities, and tasks (Cooper, 1999).

5.1.1 Using Personas to Develop Sample User CAPs

This section provides brief example personas based on interviews and other information gathering techniques that describe various users with various capabilities and other characteristics. These descriptions focus on the user's skills and background. Unlike Cooper's original approach, photographs are not included. As with Cooper's approach, all names are fictitious.

Following each description is an analysis of what input and output modality capabilities the CAP for this user would document, example CAP entries from the user's CAP, and a high level summary of the user's CAP.

5.1.2 Clark

5.1.2.1 Clark's Description

A mild-mannered reporter, Clark uses computers at work and home and has little difficulty in their use. He wears glasses. He is fluent in English.

Given the above description, Table 5.2 shows that we can create a user CAP specifically for Clark. We start by documenting his CAP_{IC}.

Table 5.2: A CAP for Clark

CAP_{IC} Identification	
Name	Clarks_Cap
Type	CAP_{USE}
Qualifier	MAY
Description	This is the starting point of Clark's user CAP.
CAP_{IC} Linkages	
Higher-CAP_O	{<Root, >}
Peer-CAP_{IC}	
Lower-CAP_{IR}	{<Clark_IR, >}
Lower-CAP_{PF}	
Lower-CAP_{OT}	{<Clark_OT, >}

Table 5.2 documents the start of Clark's CAP:

- Type is filled with the default entry for a user CAP.
- Qualifier is "MAY" to indicate the optional use of this user's CAP by the CAP_O.
- Higher-CAP_O links to the Root CAP_O specified at the beginning of the chapter.
- Peer-CAP_{IC} does not link to any peer user CAPs, since Clark's CAP does not interact with any of the other user CAPs in the CAP_O. Note that this attribute can be later used to link to the CAPs of systems and environments with which Clark interacts.

Now we need to document the various CFs that describe Clark. To achieve this, we need CAP_{CF}s that describe all of the modality, capability, and processing information that is relevant to him for each of input (CAP_{IR}), output (CAP_{OT}), and processing (CAP_{PF}).

Given the above description of this user, his CAP_{CF}s could document the following:

Speech: This user has full speech capabilities, since the user does not report anything affecting his speech capabilities.

Sight: Although he wears glasses, this user has full sight capabilities since the user does not report anything affecting his sight capabilities.

Hearing: This user has full hearing capabilities since the user does not report anything affecting his hearing capabilities.

Tactile: This user has full tactile capabilities since the user does not report anything affecting his ability to feel heat and vibration. In addition, this user has full mobility tactile capabilities since the user does not report anything affecting his mobility capabilities. This user also has full dexterity tactile capabilities since the user does not report anything affecting his dexterity capabilities.

5.1.2.2 Clark's CAP

Thus, as shown in Table 5.3, this user's CAP_{CF}s must document the capability of fully using ALL available input modalities.

Table 5.3: A CAP for Clark's Modality Input Capabilities

Identification	
Type	CAP_{IR}
Name	Clark_IR
Qualifier	MAY
Description	Clark's modality input capabilities.
Linkages	
Higher-CAP_{IC}	{<Clarks_Cap, >}
Peer-CAP_{IR}	
Peer-CAP_{PF}	
Peer-CAP_{OT}	{<Clark_OT, >}
Lower-CAP_{M}	{<Clark-AllModality, >}
Lower-CAP_{C}	
Lower-CAP_{P}	
Communication capabilities	
Capacity	N
Parallel	N
Serial	N
Degree	
Intermittent	
Devoted	

Table 5.3 documents Clark's capabilities to perceive data:

- Type indicates this is an IR record,
- Qualifier is "MAY" to indicate the optional use in this user's CAP,
- Higher-CAP_{IC} links to this user's CAP_{IC},
- Peer-CAP_{IR} has no link to any peer CAP_{IR}, since this record is the only CAP_{IR}.
- Peer-CAP_{OT} links to the peer CAP_{OT} also contained in this user's CAP, and
- Lower-CAP_{M} links to the CAP_{M} used by this CAP_{IR}.
- Communication capabilities suggest that this user supports innumerable input channels both contiguously and successively.

As mentioned in Chapter 3, the Communication capabilities attributes *Degree*, *Intermittent*, and *Devoted* are only set when a channel is being used by the CF to communicate with another

IC (e.g., a system). Since, in this chapter, the example CAPs are not connected to any channels, these three attributes are left blank. Further, since actually connecting CAPs is beyond the scope of this Thesis, these communication capabilities attributes are currently merely reserved for future use.

This user's CAP_{CF}s document the capability of fully using all available output modalities.

Table 5.4: A CAP for Clark's Modality Output Capabilities

Identification	
Type	CAP_{OT}
Name	Clark_OT
Qualifier	MAY
Description	Clark's modality output capabilities.
Linkages	
Higher-CAP_{IC}	{<Clarks_Cap, >}
Peer-CAP_{IR}	{<Clark_IR, >}
Peer-CAP_{PF}	
Peer-CAP_{OT}	
Lower-CAP_{M}	{<Clark-AllModality, >}
Lower-CAP_{C}	
Lower-CAP_{P}	
Communication capabilities	
Capacity	N
Parallel	N
Serial	N
Degree	
Intermittent	
Devoted	

Table 5.4 is a little different from the CAP_{IR} record shown in Table 5.3. The major differences are:

- Type is filled with the default entry for an OT rather than for an IR,
- Peer-CAP_{IR} links to the peer CAP_{IR} also contained in this user's CAP,
- Peer-CAP_{OT} has no link to any peer CAP_{OT}, since this record is the only CAP_{OT}, and
- Communication capabilities suggest that this user supports innumerable output channels both contiguously and successively.

Since this user is able to use all available input and output modalities, we only need to create one CAP_{M}. It will then be linked by both the user's CAP_{IR} and CAP_{OT}.

Table 5.5: Clark's Only Modality Record

Identification	
Type	CAP_M
Name	Clark-AllModality
Qualifier	MAY
Description	Clark's modalities.
Linkages	
Higher-CAP_{IR}	{<Clark_IR, >}
Higher-CAP_{OT}	{<Clark_OT, >}
Higher-CAP_{PF}	
Peer-CAP_M	
Peer-CAP_C	
Peer-CAP_P	
Modality-Specific Information	
ModalityType	`ALL`
MediaTypes	{`ALL`}
Language	{<`eng, Latn`>}

Table 5.5 documents Clark's supported input and output modalities:

- Type is filled with the default entry for a CAP_M,
- Higher-CAP_{IR} links to this user's CAP_{IR},
- Higher-CAP_{OT} links to this user's CAP_{OT},
- ModalityType is set to ALL indicating that all modalities are supported by this user,
- MediaTypes is set to ALL indicating that all media are supported by this user, and
- Language is set to show this user's language capabilities (in this case, English together with the Latin script).

Since CAP_{PF}s are only used to describe ATs, they are not required in CAP_{USE}s. Therefore, the above fully documents Clark's CAP.

At a high level, Table 5.6 describes the structure of Clark's CAP.

Table 5.6: Structure of CAP_{USE} for "Clark"

Name	**Type**	**Unary Operator**	**Tables**
1.0 Clarks_Cap	CAPUSE	MAY	Table 5.2
1.1 Clark_IR	CAPIR	MAY	Table 5.3
1.2 Clark_OT	CAPOT	MAY	Table 5.4
1.1.1 Clark-AllModality	CAPM	MAY	Table 5.5

5.1.3 Johann

5.1.3.1 Johann's Description

Well-versed in computers, Johann has little difficulty in their use. Johann wears glasses to correct mild shortsightedness. Johann is hard of hearing and has trouble hearing error tones, if the volume is not sufficiently loud. In addition, he is unable to hear in the left ear and only able to hear sounds within a specific range of volume and pitch in the right. He is fluent in German. As a side effect of being hard of hearing, his speech contains some odd inflection and nasality. Johann is dominantly left-handed and tends to prefer using a mouse in the left hand to the point where he will only tolerate using ergonomically right-handed mice in his left hand for short periods.

Given the above description, we can create a user CAP specifically for Johann. We start by documenting his CAP$_{IC}$ shown in Table 5.7.

Table 5.7: A CAP for Johann

CAP$_{IC}$ Identification	
Name	Johanns_Cap
Type	CAP$_{USE}$
Qualifier	MAY
Description	This is the starting point of Johann's user CAP.
CAP$_{IC}$ Linkages	
Higher-CAP$_{O}$	{<Root, >}
Peer-CAP$_{IC}$	
Lower-CAP$_{IR}$	{<Johann_Visual_IR, >, <Johann_Tactile_IR, >, <Johann_Auditory_IR, >,}
Lower-CAP$_{PF}$	
Lower-CAP$_{OT}$	{<Johann_OT, >}

Table 5.7 differs a little from Clark's IC shown in Table 5.2. Specifically, as will be discussed later, because this user does not use all available input modalities, the Lower-CAP$_{IR}$ links to more than one CAP, each of which document Johann's perceptual capabilities in specific modalities.

Now we need to document the various CFs that describe Johann. To achieve this, we need CAP$_{CF}$s that describe all of the modality, capability, and processing information that is relevant to Johann for each of input (CAP$_{IR}$) and output (CAP$_{OT}$).

Given the above description of this user, his CAP$_{CF}$s could document the following:

Speech: The user has full speech capabilities. While the odd inflection and nasality reported by this user may impact the use of system input options such as speech-recognition, since the user does not report any difficulties with speech-recognition software, it is not an issue for the user's CAP to record.

Sight: The user reports mild nearsightedness that is corrected with glasses. This user has full sight capabilities.

Hearing: This user reports monaural hearing capability. The user also reports hearing capability within a specific range of volume and pitch. Monaural hearing means that the user cannot be tasked with processing information relying solely on stereo hearing and that any audio input favour the ear that can hear. This should be reflected in this user's CAP.

No information is provided to determine what range of sounds may need to be amplified (i.e., high or low frequencies). This is expected, since audiometric or other medical diagnostic information is not likely to be explicitly known to all users. Two options in this case are to:

- ignore the processing needed to adjust audio output into the user's range of hearing and wait until the user explicitly specifies such an adjustment, or
- specifically encode the CAP describing their hearing using specific assumptions.

There are four ways to resolve this second option:

First, the user could be asked if they know what types of sounds they hear well. This assumes the user knows if they have difficulty hearing high frequency sounds, the most common type of sensorineural hearing loss (Nadol, 1993), and/or they have difficulty hearing low frequency sounds. While this approach provides rather poor precision, it may be just enough (or minimally sufficient) to provide better access for the user, since it can communicate to the system the need to adjust groups of sounds into perceivable ranges.

Second, if the system was to learn the user's range of hearing through system volume adjustments over time, then information on the minimum and maximum volume (i.e., comfort levels) can be collected and used to compute the user's capabilities and update (with the user's permission) the user's own CAP. This is a more adaptive approach which may lead to greater precision for amplification but, without being combined with the first approach, may have no impact on the user's perception of all auditory information, because it does not adjust the range of frequencies the system uses into a limited group which the user can perceive.

Third, the system could provide some pure tone hearing test to determine the hearing capabilities of the user. While the precision of such a test is strongly influenced by a combination of hardware and environment, the results could provide a system with a strong starting point from which to further apply an adaptive approach, such as the second option. Note that this approach may be culturally inappropriate or may take too long for the user's comfort.

Fourth, do nothing about frequency range or amplification, since actions based on the wrong assumption(s) will provide no benefit to the user. This is the least preferred approach.

Tactile: This user has full tactile capabilities, since there is no report of anything affecting his ability to feel heat and vibration. For the same reason, this user has full mobility tactile capabilities too. As well,, this user also has full dexterity tactile capabilities.

The user reports use of ergonomically right-hand-specific devices is difficult and/or uncomfortable due to being dominantly left-handed. Although ergonomically left-handed mice are available, current system support for left-handed users is focused only in setting the primary mouse button of a multi-button mouse to the right-most button.

However, the need for left-handed orientation is a preference. The user is not unable to use their right hand or even to use right-handed input devices; the user is only inconvenienced by them. A preference is not an access issue but an accessibility issue in terms of usability. The CAP, as it is currently implemented, focuses only on whether access is possible.

5.1.3.2 Johann's CAP

This user's CAP_{CF}s document the capability of using the most available input modalities and their monaural hearing capability. The correct approach to this is to separately and individually document every one of the user's input capabilities. This approach adheres to the CAP_{USE} Production Rules above.

First, we document that this user is capable of fully using visual modality input. Table 5.8 shows the CAP_{IR}.

Table 5.8: Johann's Visual Modality Input

Identification	
Type	CAP_{IR}
Name	Johann_Visual_IR
Qualifier	MAY
Description	Johann's visual modality input capabilities.
Linkages	
Higher-CAP_{IC}	{<Johanns_Cap, >}
Peer-CAP_{IR}	{<Johann_Tactile_IR, >, Johann_Auditory_IR, >,}
Peer-CAP_{PF}	
Peer-CAP_{OT}	{<Johann_OT, >}
Lower-CAP_{M}	{<Johann-Visual_Input_Modality, >}
Lower-CAP_{C}	
Lower-CAP_{P}	
Communication capabilities	
Capacity	N
Parallel	N
Serial	N
Degree	
Intermittent	
Devoted	

Table 5.8 documents Johann's capabilities to perceive visual data:

- Peer-CAP$_{IR}$ lists the other IR records contained in Johann's CAP, because each of these CAPs interacts with the others.
- Peer-CAP$_{OT}$ lists the only OT record contained in Johann's CAP, because this CAP interacts with each of the CAP$_{IR}$s.
- Lower-CAP$_M$ links to a CAP$_M$ describing Johann's visual input modality, media, and languages.

The corresponding CAP$_M$ appears in Table 5.9.

Table 5.9: Johann's Visual Modality Input Media

Identification	
Type	CAP$_M$
Name	Johann-Visual_Input_Modality
Qualifier	MAY
Description	Johann can use any visual input media.
Linkages	
Higher-CAP$_{IR}$	{<Johann_Visual_IR, >}
Higher-CAP$_{OT}$	
Higher-CAP$_{PF}$	
Peer-CAP$_M$	
Peer-CAP$_C$	
Peer-CAP$_P$	
Modality-Specific Information	
ModalityType	VISUAL
MediaTypes	{`TextWritten`, `Picture`, `VisualModel`, `Movie`, `DynamicVisualModel`, `Gesture`}
Language	{<`deu, Latn`>}

Table 5.9 documents the types of media and languages Johann can use in the visual modality:

- MediaTypes lists the types of media that are specific to the visual modality.
- Language is set to show this user's language capabilities (in this case, German together with the Latin script). This attribute, together with the previous attribute's documentation of support for TextWritten, informs us that Johann can read German text.

Documenting this user's capability to fully use tactile modality inputs includes a similar pair of CAP$_{IR}$ and CAP$_M$ records for the tactile modality.

Documenting this user's auditory input capabilities is similar; however, it also requires the addition of a CAP$_C$. Table 5.10 shows the IR record.

Table 5.10: Johann's Auditory Modality Input

Identification	
Type	CAP$_{IR}$
Name	Johann_Auditory_IR
Qualifier	MAY
Description	Johann's auditory modality input capabilities.
Linkages	
Higher-CAP$_{IC}$	{<Johanns_Cap, >}
Peer-CAP$_{IR}$	{<Johann_Tactile_IR, >, <Johann_Visual_IR, >,}
Peer-CAP$_{PF}$	
Peer-CAP$_{OT}$	{<Johann_OT, >}
Lower-CAP$_{M}$	{<Johann-Auditory_Input_Modality, >}
Lower-CAP$_{C}$	{<Johann-Auditory_Input_Capability, >}
Lower-CAP$_{P}$	
Communication capabilities	
Capacity	N
Parallel	N
Serial	N
Degree	
Intermittent	
Devoted	

Table 5.10 documents this user's ability to perceive auditory data:

- Lower-CAP$_{C}$ links to a CAP$_{C}$ to list the capabilities of the user's auditory modality input.

Table 5.11 shows the corresponding CAP$_{M}$.

Table 5.11: Johann's Auditory Modality Input Media

Identification	
Type	CAP$_{M}$
Name	Johann-Auditory_Input_Modality
Qualifier	MAY
Description	Johann can use certain auditory input.
Linkages	
Higher-CAP$_{IR}$	{<Johann_Auditory_IR, >}
Higher-CAP$_{OT}$	
Higher-CAP$_{PF}$	
Peer-CAP$_{M}$	
Peer-CAP$_{C}$	{<Johann-Auditory_Input_Capability, >}
Peer-CAP$_{P}$	
Modality-Specific Information	
ModalityType	AUDITORY
MediaTypes	{`TextSpoken`, `Sound`, `Music`}
Language	{<`deu`, `NIL`>}

Table 5.11 documents this user's ability to use auditory data:

- MediaTypes lists the types of media that are specific to the auditory modality.
- Language is set to show this user's language capabilities (in this case, German). Note that there is no known script to document German in the auditory modality.

Table 5.12 indicates this user's unique auditory modality input capabilities in terms of a CAP$_C$.

Table 5.12: Johann's Auditory Modality Input Capabilities

Identification	
Type	CAP$_C$
Name	Johann-Auditory_Input_Capability
Qualifier	MAY
Description	This capability record documents monaural auditory modality input capabilities.
Linkages	
Higher-CAP$_{IR}$	{<Johann_Auditory_IR, >}
Higher-CAP$_{OT}$	
Higher-CAP$_{PF}$	
Peer-CAP$_M$	{<Johann-Auditory_Input_Modality, >}
Peer-CAP$_C$	
Peer-CAP$_P$	
Capability-Specific Information	
Capability name	Monaural
Capability instance	0
Capability values	Right

The CAP$_C$ in Table 5.12 documents:

- Capability name "Monaural" enables users to state that of the two or more channels of stereo audio output a system may use, only one can be heard.
- Observe that no provision is made for specific amplification or minimum volume.

This user has monaural hearing capability and does not have stereo hearing capability. Information encoded in stereo sound cannot be decoded by this user. Table 5.12 notes this user capability by ensuring all audio output favours the user's right ear.

This user's CAP$_{CF}$s also document the capability of fully using all available output modalities. With the exception of the language setting (i.e., <deu, Latn> is used to document support for German), this is achieved in the same manner as Clark's CAP above and will not be repeated here.

The above fully documents Johann's CAP. At a high level, Table 5.13 describes the structure of Johann's CAP.

Table 5.13: Structure of CAP_{USE} for "Johann"

Name	Type	Unary Operator	Tables
1.0 Johanns_Cap	CAPUSE	MAY	Table 5.7
1.1 Johann_Tactile_IR	CAPIR	MAY	Like Table 5.8
1.1.1 Johann-Tactile_Input_Modality	CAPM	MAY	Like Table 5.9
1.2 Johann_Visual_IR	CAPIR	MAY	Table 5.8
1.2.1 Johann-Visual_Input_Modality	CAPM	MAY	Table 5.9
1.3 Johann_Auditory_IR	CAPIR	MAY	Table 5.10
1.3.1 Johann-Auditory_Input_Modality	CAPM	MAY	Table 5.11
1.3.2 Johann-Auditory_Input_Capability	CAPC	MAY	Table 5.12
1.4 Johann_OT	CAPOT	MAY	Like Table 5.4
1.4.1 Johann-AllModality	CAPM	MAY	Like Table 5.5

5.1.4 Pam

5.1.4.1 Pam's Description

Highly computer literate, Pam reports that, ever since she had a home automation system installed, she uses her computer for daily living. Pam has been quadriplegic, since a car accident several years ago, and is currently only able to move her head within a limited range due to restrictive neck support. She reports sensation in her shoulders and above only. She uses a respirator to support breathing, which regularly interrupts her speech. When using a desktop computer, she prefers speech recognition and dictation software but occasionally uses an onscreen keyboard in select mode with word prediction software. For mousing, she currently uses a HeadMouse.

Given the above description, we can create a user CAP specifically for Pam. We start by documenting her CAP_{IC}, shown in Table 5.14.

Table 5.14: A CAP for Pam

CAP_{IC} Identification	
Name	Pams_Cap
Type	CAP_{USE}
Qualifier	MAY
Description	This is the starting point of Pam's user CAP.
CAP_{IC} Linkages	
Higher-CAP_O	{<Root, >}
Peer-CAP_{IC}	
Lower-CAP_{IR}	{<Pam-Tactile_IR, >, <Pam-Visual_IR, >, <Pam-Auditory_IR, >}
Lower-CAP_{PF}	
Lower-CAP_{OT}	{<Pam-Auditory_OT, >, <Pam-Tactile_OT, >}

To document the various Component Features that describe Pam, we need CAP_{CF}s that describe all of the modality, capability, and processing information that is relevant to Pam for each of input (CAP_{IR}) and output (CAP_{OT}).

Given the above description of this user, her CAP_{CF}s could document the following:

Speech: This user has full speech capabilities. However, the user reports that her breathing is supported by the use of respirator equipment and that, as a side effect, flow of speech is regularly interrupted by the respirator system forcing the user to breathe. This may impact the use of speech-to-text and speech-recognition system input options. However, since the user does not report any difficulties with speech-recognition software, it is not an issue for the user's CAP to record.

Sight: This user has full sight capabilities, since there is no report of anything affecting her sight capabilities.

Hearing: This user has full hearing capabilities, since there is no report of anything affecting her hearing capabilities.

Tactile: The user reports that her ability to feel heat and vibration is restricted to the shoulders, neck and higher. This impacts the selection of system tactile/haptic output options.

In addition, this user reports that use of all limbs is not possible and *gross* body movement is restricted to the head and neck. The user also reports limited range of head and neck movement due to restrictive neck support. This impacts the selection of system input or output options requiring gross motor skills tactile capabilities such as reach. For many similar users, current system support is primarily focused on head-based control (e.g., HeadMouse, sip/puff control, head stick, etc).

The user also reports that *fine* body movement is restricted to the head and neck. Thus, no extremities (e.g., fingers, toes) can be moved. The user has fine-motor control tactile capabilities with the head and neck. This will impact the selection of system input or output options requiring fine motor skills tactile capabilities. For many similar users, current system support is primarily focused on head-based control. Augmenting such controls with an onscreen keyboard and word prediction software can substitute for a traditional keyboard.

5.1.4.2 Pam's CAP

This user's CAP_{CF}s document the capability of using specific input modalities. In this case, full hearing, full sight, tactile sensation (i.e., using specific body loci and modes of stimulation) and motor control above the shoulders capabilities are documented in Pam's CAP. Every one of the user's input capabilities need to be specified. This approach adheres to the CAP_{USE} Production Rules above.

As shown in Table 5.15, Pam's tactile input capabilities can be described in a manner similar to that of Johann's.

Table 5.15: Pam's Tactile Modality Input

Identification	
Type	CAP$_{IR}$
Name	Pam-Tactile_IR
Qualifier	MAY
Description	Pam's tactile modality input capabilities.
Linkages	
Higher-CAP$_{IC}$	{<Pams_Cap, >}
Peer-CAP$_{IR}$	{<Pam-Visual_IR, >, <Pam-Auditory_IR, >}
Peer-CAP$_{PF}$	
Peer-CAP$_{OT}$	{<Pam-Auditory_OT, >, <Pam-Tactile_OT, >, <Pam-Visual_OT>}
Lower-CAP$_{M}$	{<Pam-Tactile_Input_Modality, >}
Lower-CAP$_{C}$	{<Pam-Tactile_Input_Capability, >}
Lower-CAP$_{P}$	
Communication capabilities	
Capacity	N
Parallel	N
Serial	N
Degree	
Intermittent	
Devoted	

Table 5.16 shows the CAP$_{M}$ for her tactile input capabilities.

Table 5.16: Pam's Tactile Modality Input Media

Identification	
Type	CAP$_{M}$
Name	Pam-Tactile_Input_Modality
Qualifier	MAY
Description	Pam can use tactile input.
Linkages	
Higher-CAP$_{IR}$	{<Pam-Tactile_IR, >}
Higher-CAP$_{OT}$	
Higher-CAP$_{PF}$	
Peer-CAP$_{M}$	
Peer-CAP$_{C}$	{<Pam-Tactile_Input_Capability, >}
Peer-CAP$_{P}$	
Modality-Specific Information	
ModalityType	TACTILE
MediaTypes	{`Temperature`, `Texture`, `TactileGraphic`, `ForceFeedback`}
Language	{<`NONE`, `NIL`>}

Table 5.16 documents this user's ability to perceive tactile data:

- ModalityType is set to "TACTILE" to show that this user supports tactile modalities,
- MediaTypes is set to indicate that this user supports tactile media (except tactile text), and
- Language is set to show that, for the tactile modality, this user is language neutral.

Pam's range of motion capabilities and the small number of available loci of stimulation (she reports feeling above her shoulders), mean that any system's tactile output must be specific to these areas and require little adjustment of tactile devices for independent use. Thus, as shown in Table 5.17, her CAP documents that she has tactile capabilities on her head, neck, and shoulders.

Table 5.17: Pam's Tactile Modality Input Capabilities

Identification	
Type	CAP_C
Name	Pam-Tactile_Input_Capability
Qualifier	MAY
Description	The capabilities of Pam's tactile capabilities.
Linkages	
Higher-CAP_{IR}	{<Pam-Tactile_IR, >}
Higher-CAP_{OT}	
Higher-CAP_{PF}	
Peer-CAP_M	{<Pam-Tactile_Input_Modality, >}
Peer-CAP_C	
Peer-CAP_P	
Capability-Specific Information	
Capability name	Head
Capability instance	0
Capability values	
Capability name	Neck
Capability instance	0
Capability values	
Capability name	Shoulders
Capability instance	0
Capability values	

Table 5.17 documents this user's ability to use tactile data:

- Capability name lists the appropriate loci for tactile stimulation, and
- Capability values may list the levels of stimulation for each loci. Here, capability values are not provided; thus, a tactile device can use its own default settings until otherwise specified.

Documenting Pam's visual input capabilities is achieved in the same manner as Johann's visual input capabilities (with the exception Language is set to <eng, Latn> to show that she can read English), and will not be repeated here.

Because Pam has full auditory input capabilities (i.e., hearing), documenting them is approached differently from Johann's CAP. The CAP$_{IR}$ is shown in Table 5.18.

Table 5.18: Pam's Auditory Modality Input

Identification	
Type	CAP$_{IR}$
Name	Pam-Auditory_IR
Qualifier	MAY
Description	Pam's auditory modality input capabilities.
Linkages	
Higher-CAP$_{IC}$	{<Pams_Cap, >}
Peer-CAP$_{IR}$	{<Pam-Tactile_IR, >, <Pam-Visual_IR, >}
Peer-CAP$_{PF}$	
Peer-CAP$_{OT}$	{<Pam-Auditory_OT, >, <Pam-Tactile_OT, >}
Lower-CAP$_{M}$	{<Pam-Auditory_Input_Modality, >}
Lower-CAP$_{C}$	
Lower-CAP$_{P}$	
Communication capabilities	
Capacity	N
Parallel	N
Serial	N
Degree	
Intermittent	
Devoted	

Then, to fully document this IR, we add the CAP$_{M}$ shown in Table 5.19.

Table 5.19: Pam's Auditory Modality Input Media

Identification	
Type	CAP$_{M}$
Name	Pam-Auditory_Input_Modality
Qualifier	MAY
Description	Modality record for Pam's auditory input capabilities.
Linkages	
Higher-CAP$_{IR}$	{<Pam-Auditory_IR, >}
Higher-CAP$_{OT}$	
Higher-CAP$_{PF}$	
Peer-CAP$_{M}$	
Peer-CAP$_{C}$	
Peer-CAP$_{P}$	
Modality-Specific Information	
ModalityType	AUDITORY
MediaTypes	{`TextSpoken`, `Sound`, `Music`}
Language	{<`eng`, `NIL`>}

The above fully documents this user's input capabilities.

This user's CAP_{CF}s also document the capability of using specific output modalities. In this case, Pam's auditory (i.e., speech), and tactile (i.e., mobility and dexterity of the head and neck) output capabilities are documented.

As with her input capabilities, we separately document every one of the user's output capabilities including the user's unique output capabilities. This approach adheres to the CAP_{USE} Production Rules above.

We will start with the user's auditory output capabilities. Table 5.20 documents that Pam can produce auditory output.

Table 5.20: Pam's Auditory Modality Output

Identification	
Type	CAP_{OT}
Name	Pam-Auditory_OT
Qualifier	MAY
Description	Pam's auditory modality output capabilities.
Linkages	
Higher-CAP_{IC}	{<Pams_Cap, >}
Peer-CAP_{IR}	{<Pam-Tactile_IR, >, <Pam-Visual_IR, >, <Pam-Auditory_IR, >}
Peer-CAP_{PF}	
Peer-CAP_{OT}	{<Pam-Tactile_OT, >}
Lower-CAP_{M}	{<Pam-Auditory_Output_Modality, >}
Lower-CAP_{C}	{<Pam-VoiceRecognition, >}
Lower-CAP_{P}	
Communication capabilities	
Capacity	N
Parallel	N
Serial	N
Degree	
Intermittent	
Devoted	

Table 5.20 documents this user's ability to produce auditory data:

- Type is filled with the default entry for a CAP_{OT} record,
- Qualifier is "MAY" to indicate that this user can use speech-based system input options such as speech recognition, if available,
- Higher-CAP_{IC} links to this user's CAP_{IC},
- Peer-CAP_{OT} links to the other output modalities documented in this user's CAP,
- Lower-CAP_{M} links to the CAP_{M}s further documenting the auditory output capabilities of this user, and

- Lower-CAP$_C$ links to the CAP$_C$s further documenting the capabilities of this user.

To fully document this OT, information on the modality of use and capabilities needs to be documented. The CAP$_M$ is shown in Table 5.21.

Table 5.21: Pam's Auditory Modality Output Media

Identification	
Type	CAP$_M$
Name	Pam-Auditory_Output_Modality
Qualifier	MAY
Description	Pam's auditory output capability modality record.
Linkages	
Higher-CAP$_{IR}$	
Higher-CAP$_{OT}$	{<Pam-Auditory_OT, >}
Higher-CAP$_{PF}$	
Peer-CAP$_M$	
Peer-CAP$_C$	{<Pam-VoiceRecognition, >}
Peer-CAP$_P$	
Modality-Specific Information	
ModalityType	AUDITORY
MediaTypes	{`TextSpoken`, `Sound`, `Music`}
Language	{<`eng`, `NIL`>}

Table 5.21 documents the kinds of auditory data this user can produce:

- MediaTypes include both sound and music, since one can make general sounds or sing as well as speak, and
- Language is set to reflect that the user speaks English and that there is no known auditory modality based writing system for this language.

This user reports that she "prefers" speech recognition and dictation software, but occasionally uses an onscreen keyboard with word prediction software. In this case, her "preference" is actually a statement of capability that she can use speech recognition and dictation software, as well as an onscreen keyboard and word prediction software. She needs the system to support either speech recognition and dictation software or an onscreen keyboard and word prediction software to have full access. If in fact a system happens to support *both* options, she can choose which to use or combine all of them.

The CAP$_C$ shown in Table 5.22 documents the user's capabilities to provide speech output to be used by an IC capable of voice recognition. Note that the CAP does not specifically describe her speech patterns, only her speech capabilities. To describe her speech patterns a voice recognition profile, a file that encodes characteristics of the user's voice to improve the accuracy of recognition, should be used. Such files result from training voice recognition software and can be communicated separately to a system along with a user's CAP.

Table 5.22: Pam's Auditory Modality Output Capabilities

Identification	
Type	CAP_C
Name	Pam-VoiceRecognition
Qualifier	MAY
Description	This capability configures voice recognition.
Linkages	
Higher-CAP_{IR}	
Higher-CAP_{OT}	{<Pam-Auditory_OT, >}
Higher-CAP_{PF}	
Peer-CAP_M	{<Pam-Auditory_Output_Modality, >}
Peer-CAP_C	
Peer-CAP_P	
Capability-Specific Information	
Capability name	Intensity
Capability instance	0
Capability values	

Table 5.22 documents this user's ability to output speech data:

- Capability values may list how loudly or softly (i.e., Intensity) the user speaks. This information assists the system in determining microphone gain. In this example, capability values are not provided; thus, a microphone can use its own default settings until otherwise specified.

A voice recognition system can be used for mouse control as well as dictation. Pam uses a head-mounted mouse and may use some other mouse emulation rather than full speech-based control. If a system does not support a HeadMouse, or other mouse emulation she can use, it should ask whether to use the voice recognition system for mouse control. Let us now move this discussion to her tactile output capabilities.

When coupled with a traditional desktop computer system, a user's tactile output capabilities often relate primarily to keyboarding and mousing. Such capabilities can also be related to non-traditional system input devices, such as sip/puff switches and headmice. These tactile devices rely on either the Texture or Force Feedback media types. These media types include motor actions like movement of specific body parts (e.g., the hand) and applying some type of pressure to activate a switch (e.g., a puff of air). For Pam, her keyboarding and mousing capabilities are technically the same thing as she reports using mouse emulation in combination with the onscreen keyboard. Table 5.23 shows the CAP_{OT} for Pam's tactile output capabilities.

Table 5.23: Pam's Tactile Modality Output

Identification	
Type	CAP$_{OT}$
Name	Pam-Tactile_OT
Qualifier	MAY
Description	Pam's tactile output capabilities.
Linkages	
Higher-CAP$_{IC}$	{<Pams_Cap, >}
Peer-CAP$_{IR}$	{<Pam-Tactile_IR, >, <Pam-Visual_IR, >, <Pam-Auditory_IR, >}
Peer-CAP$_{PF}$	
Peer-CAP$_{OT}$	{<Pam-Auditory_OT, >}
Lower-CAP$_{M}$	{<Pam-Tactile_Output_Modality, >}
Lower-CAP$_{C}$	{<Pam-Tactile_Output_Capability, >}
Lower-CAP$_{P}$	
Communication capabilities	
Capacity	N
Parallel	N
Serial	N
Degree	
Intermittent	
Devoted	

The associated CAP$_{M}$ is shown in Table 5.24.

Table 5.24: Pam's Tactile Modality Output Media

Identification	
Type	CAP$_{M}$
Name	Pam-Tactile_Output_Modality
Qualifier	MAY
Description	Pam's tactile output capability modality.
Linkages	
Higher-CAP$_{IR}$	
Higher-CAP$_{OT}$	{<Pam-Tactile_OT, >}
Higher-CAP$_{PF}$	
Peer-CAP$_{M}$	
Peer-CAP$_{C}$	{<Pam-Tactile_Output_Capability, >}
Peer-CAP$_{P}$	
Modality-Specific Information	
ModalityType	TACTILE
MediaTypes	{`Texture, ForceFeedback`}
Language	{<`eng, Latn`>}

The CAP$_{C}$ describing her tactile output capabilities is similar to that of her tactile input capabilities above. The modality of a traditional keyboard is tactile. The modality of an onscreen keyboard is visual but is used through Pam's tactile output capabilities (i.e., head and neck proprioception). These CAPs document the modality of Pam's keyboarding capability.

There is no CAP_{OT} for the visual modality because currently nothing supporting communication through the eye alone exists. Users relying on eye-blinks would use a CAP_{OT} for the tactile modality.

The above fully documents Pam's CAP. Table 5.25 describes its structure.

Table 5.25: Structure of CAP_{USE} for "Pam"

Name	**Type**	**Unary Operator**	**Tables**
1.0 Pams_Cap	CAPUSE	MAY	Table 5.14
1.1 Pam-Tactile_IR	CAPIR	MAY	Table 5.15
1.1.1 Pam-Tactile_Input_Modality	CAPM	MAY	Table 5.16
1.1.2 Pam-Tactile_Input_Capability	CAPC	MAY	Table 5.17
1.2 Pam-Visual_IR	CAPIR	MAY	Like Table 5.8
1.2.1 Pam-Visual_Input_Modality	CAPM	MAY	Like Table 5.9
1.3 Pam-Auditory_IR	CAPIR	MAY	Table 5.18
1.3.1 Pam-Auditory_Input_Modality	CAPM	MAY	Table 5.19
1.4 Pam-Tactile_OT	CAPOT	MAY	Table 5.23
1.4.1 Pam-Tactile_Output_Modality	CAPM	MAY	Table 5.24
1.4.2 Pam-Tactile_Output_Capability	CAPC	MAY	Like Table 5.17
1.5 Pam-Auditory_OT	CAPOT	MAY	Table 5.20
1.5.1 Pam-Auditory_Output_Modality	CAPM	MAY	Table 5.21
1.5.2 Pam-VoiceRecognition	CAPC	MAY	Table 5.22

5.1.5 Tae

5.1.5.1 Tae's Description

Tae uses computers more often now. She has been legally blind for five years and reports that she is able to see magnified high-contrast text and use the mouse with a magnified high-contrast pointer icon. Her current system support is focused on screen magnification and high contrast settings. She switches between visual and text-to-speech output.

We can create a user CAP for Tae. We start by documenting her CAP_{IC}, shown in Table 5.26.

Table 5.26: A CAP for Tae

CAP_{IC} Identification	
Name	Taes_Cap
Type	CAP_{USE}
Qualifier	MAY
Description	This is the top of Tae's user CAP.
CAP_{IC} Linkages	
Higher-CAP_O	{<Root, >}
Peer-CAP_{IC}	
Lower-CAP_{IR}	{<Tae_Visual_IR, >, <Tae_Auditory_IR, >, <Tae_Tactile_IR, >}
Lower-CAP_{PF}	
Lower-CAP_{OT}	{<Tae_OT, >}

Now we need to document the various CFs that describe Tae. To achieve this, we need CAP_{CF}s that describe all of the modality, capability, and processing information that is relevant to Tae for each of input (CAP_{IR}) and output (CAP_{OT}).

Given the above description of this user, her CAP_{CF}s could document the following:

Speech: This user has full speech capabilities, since there is no report of anything affecting her speech capabilities.

Sight: This user reports vision loss not fully correctable with glasses. To be seen it must be magnified and presented with high-contrast colours. This impacts selection of visual system output options and must be reflected in her CAP.

Hearing: This user has full hearing capabilities, since the user does not report anything affecting her hearing capabilities.

Tactile: This user has full tactile capabilities, since the user does not report anything affecting her tactile/haptic capabilities.

This user has full mobility tactile capabilities, since the user does not report anything affecting her mobility (gross-body movement) tactile capabilities. Although, the user's low vision may impact orientation and manoeuvring within space capabilities, there is no impact on gross-motor skills such as reach.

This user has full dexterity tactile capabilities, since the user does not report anything affecting her dexterity (fine motor skills) tactile capabilities.

5.1.5.2 Tae's CAP

This user's CAP_{CF}s document the capability of fully using all available input modalities and specifically document the user's unique visual modality capabilities. The correct way to document this is to separately document every one of the user's input capabilities, including the user's unique visual modality input capabilities. This approach adheres to the CAP_{USE} Production Rules above.

To begin, we document the tactile input modality capabilities that this user is fully capable of using. This is largely accomplished in the same manner as with Johann's CAP by defining a CAP_{IR} record for the tactile modality.

Now we can document this user's unique visual modality capabilities. This is initially approached similar to Johann's auditory input modality CAP, where one documents the user's ability to perceive and use the modality's input, and then records the unique capabilities in a CAP_{C}. Thus, the relevant CAP_{IR} and CAP_{M} records are shown in Table 5.27 and Table 5.28.

Table 5.27: Tae's Visual Modality Input

Identification	
Type	CAP$_{IR}$
Name	Tae_Visual_IR
Qualifier	MAY
Description	Tae's visual modality input capabilities.
Linkages	
Higher-CAP$_{IC}$	{<Taes_Cap, >}
Peer-CAP$_{IR}$	{<Tae_Tactile_IR, >, <Tae_Visual_IR, >,}
Peer-CAP$_{PF}$	
Peer-CAP$_{OT}$	{<Tae_OT, >}
Lower-CAP$_{M}$	{<Tae-Visual_Input_Modality, >}
Lower-CAP$_{C}$	{<Tae-Colour_Capability, >, <Tae-Magnification_Capability, >}
Lower-CAP$_{P}$	
Communication capabilities	
Capacity	N
Parallel	N
Serial	N
Degree	
Intermittent	
Devoted	

Table 5.27 documents this user's ability to perceive visual data. Lower-CAP$_{C}$ links to CAP$_{C}$s listing the user's visual modality input capabilities. Table 5.28 shows the corresponding CAP$_{M}$.

Table 5.28: Tae's Visual Modality Input Media

Identification	
Type	CAP$_{M}$
Name	Tae-Visual_Input_Modality
Qualifier	MAY
Description	Tae can use certain visual input.
Linkages	
Higher-CAP$_{IR}$	{<Tae_Visual_IR, >}
Higher-CAP$_{OT}$	
Higher-CAP$_{PF}$	
Peer-CAP$_{M}$	
Peer-CAP$_{C}$	{<Tae-Colour_Capability, >, <Tae-Magnification_Capability, >}
Peer-CAP$_{P}$	
Modality-Specific Information	
ModalityType	VISUAL
MediaTypes	{`TextWritten`, `Picture`, `VisualModel`, `Movie`, `DynamicVisualModel`, `Gesture`}
Language	{<`eng, Latn`>}

Table 5.28 documents this user's ability to use visual data:

- MediaTypes lists the types of media that are specific to the visual modality.

Tae reports that she can see something if it is magnified and presented in high-contrast. This can be documented through CAP_Cs. In the case of this user, we will show how multiple CAP_Cs can be used instead of combining all Capabilities into one CAP_C, as shown within the CAPs for Johann and Pam.

Typically, a system in high contrast mode will map all colours to a single pair of foreground and background colours (e.g., monochrome white on black display). CAPs can document this capability by using colour-specific capabilities to specify the colours the user is capable of seeing (or, as in this case, needs to see). Colour capabilities can be specified in terms of frequencies, the standard RGB (Red, Green, Blue) model (sRGB), or the Taxonomy of Capabilities described in Chapter 3. An example of specifying colours based on frequency was shown in Chapter 3; however, it is not appropriate for use with CAP_{USE}s. Using the sRGB model as a basis for the Taxonomy of Capabilities allows the specification of colours and tones in terms of standardized colour names, such as white, black, green, and so on (see Stokes, Anderson, Chandrasekar, & Motta, 1996; International Electrotechnical Commission, 1999; Raggett, Hors, & Jacobs, 1999).

Table 5.29 is an example of specifying colours using standardized colour names.

Table 5.29: Tae's Monochrome Visual Modality Input Capabilities

Identification	
Type	CAP_C
Name	Tae-Colour_Capability
Qualifier	MAY
Description	This capability denotes Tae's need for monochromatic high-contrast display.
Linkages	
Higher-CAP_{IR}	{<Tae_Visual_IR, >}
Higher-CAP_{OT}	
Higher-CAP_{PF}	
Peer-CAP_M	{<Tae-Visual_Input_Modality, >}
Peer-CAP_C	{<Tae-Magnification_Capability, >}
Peer-CAP_P	
Capability-Specific Information	
Capability name	Background
Capability instance	0
Capability values	Black
Capability name	Foreground
Capability instance	0
Capability values	White

The best way to document a user's magnification capabilities in a CAP_M is to document the desired level of magnification using SI units, which is shown in Table 5.30.

Table 5.30: Tae's Magnification Capabilities for Visual Modality Input

Identification	
Type	CAP_C
Name	Tae-Magnification_Capability
Qualifier	MAY
Description	This capability denotes Tae's need for display magnification.
Linkages	
Higher-CAP_{IR}	{<Tae_Visual_IR, >}
Higher-CAP_{OT}	
Higher-CAP_{PF}	
Peer-CAP_M	{<Tae-Visual_Input_Modality, >}
Peer-CAP_C	{<Tae-Colour_Capability, >}
Peer-CAP_P	
Capability-Specific Information	
Capability name	MagnificationLevel
Capability instance	0
Capability values	{<300, 300, %>}

In this record, the capability MagnificationLevel is given the value 300%. This means that the user reports a need for screen magnification to triple the size of screen elements. Percent is an accepted SI unit (United States Department of Commerce National Institute of Standards and Technology [NIST], 1995).

Tae also has specific mousing capabilities such as cursor size and high contrast cursor display. These are already documented by the colour and magnification CAP_Cs shown above.

Documenting the user's auditory capabilities is accomplished in the same manner as with Pam's CAP by defining CAP_{IR} and CAP_M records for the auditory modality.

Tae reports that she likes to use text-to-speech output. This "preference" is in fact a capability to use speech input, which the user can access via a screen reader, if it is available.

Research suggests (e.g., Asakawa, Takagi, Ino, & Ifukube, 2003) that expert blind users are capable of understanding speech up to 500 words per minute (wpm).

Screen readers are able to support a range of speech pitches. A speech pitch approaching 0 is a flat, monotonic voice. A speech pitch around 50 is normal inflection. A speech pitch greater than 50 is an animated voice. Furthermore, how the system renders speech pitch, and thus use the speech pitch settings, is dependent on the voice families it supports (e.g., male, female, child, etc.).

For this reason, a CAP_C can be added to her auditory input modality CAP to document screen reader capabilities. This is shown in Table 5.31.

Table 5.31: Tae's Speech-Based Auditory Modality Input Capabilities

Identification	
Type	CAP$_{C}$
Name	Tae-Speech_Capability
Qualifier	MAY
Description	Tae's speech auditory input capabilities.
Linkages	
Higher-CAP$_{IR}$	{<Tae_Auditory_IR, >}
Higher-CAP$_{OT}$	
Higher-CAP$_{PF}$	
Peer-CAP$_{M}$	{<Tae-Auditory_Input_Modality, >}
Peer-CAP$_{C}$	
Peer-CAP$_{P}$	
Capability-Specific Information	
Capability name	SpeechRate
Capability instance	0
Capability values	<200, 300, wpm>
Capability name	SpeechPitch
Capability instance	0
Capability values	50
Capability name	SpeechIntensity
Capability instance	0
Capability values	<100, 100, %>

Table 5.31 documents this user's ability to receive speech data:

- Capability values may list how fast (i.e., SpeechRate), high/low (i.e., SpeechPitch), and loud/soft (i.e., SpeechIntensity) the user is capable of understanding and using speech output. SpeechIntensity is defined in terms of percent of current system volume.

This user's CAP$_{CF}$s also document the capability of fully using all available output modalities. This is achieved in the same manner as Clark's CAP above and will not be repeated here.

This completes the discussion of Tae's CAP. At a high level, Table 5.32 describes the structure of Tae's CAP.

Table 5.32: Structure of CAP_{USE} for "Tae"

Name	Type	Unary Operator	Tables
1.0 Taes_Cap	CAPUSE	MAY	Table 5.26
1.1 Tae_OT	CAPOT	MAY	Like Table 5.4
1.1.1 Tae-AllModality	CAPM	MAY	Like Table 5.5
1.1 Tae_Tactile_IR	CAPIR	MAY	Like Table 5.15
1.1.1 Tae-Tactile_Input_Modality	CAPM	MAY	Like Table 5.16
1.2 Tae_Auditory_IR	CAPIR	MAY	Like Table 5.18
1.2.1 Tae-Auditory_Input_Modality	CAPM	MAY	Like Table 5.19
1.3.4 Tae-Speech_Capability	CAPC	MAY	Table 5.31
1.3 Tae_Visual_IR	CAPIR	MAY	Table 5.27
1.3.1 Tae-Visual_Input_Modality	CAPM	MAY	Table 5.28
1.3.2 Tae-Colour_Capability	CAPC	MAY	Table 5.29
1.3.3 Tae-Magnification_Capability	CAPC	MAY	Table 5.30

5.1.6 Summary

These example user CAPs were, by necessity, brief. A user's CAP can also have records to document several other capabilities. This ends the discussion on example CAPs for users. Now we explore developing example CAPs for systems.

5.2 Systems

The following discussion lists various systems that CAPs could describe. These descriptions are not always complete and detailed. CAPs, when used to describe systems, users, and environments can be highly detailed.

The following Production Rules are to be applied when creating CAP_{SYS}s:

1. Approach CAP_{SYS}s as an additive model. Always assume no support for any modalities and media types, unless otherwise stated.

2. System capabilities are documented in terms of the supporting hardware and software available.

3. To the extent possible, CAP_{SYS}s should be based on manufacturer's specifications. Even generic CAP_{SYS}s should have some basis in reality.

4. CAP_{SYS} names can focus on modalities as well as systems and their components.

5. CAP_{SYS}s must use the data values described in Section 3.2.

6. Because a preference is not an access issue, CAP_{SYS}s cannot be used to document preferences.

Note that there are two ways to document a system within a CAP:

1. To document each individual system at the CAP_{IC} level (i.e., treat the top level of the system as a CAP_{SYS}) and document all system components including peripherals and software, at the CAP_{IR} and CAP_{OT} level, as inputs and outputs of the system. For example, a keyboard is documented in a CAP_{IR} as a tactile modality input. This approach means that devices that have both input and output properties (e.g., a force feedback joystick) would be documented under separate CAP_{IR} and CAP_{OT} records.

2. To document all system components including software and peripherals in terms of CAP_{SYS} records. For example, a keyboard is documented in a CAP_{SYS} as a system component. This approach has the advantage of allowing devices that have both input and output properties (e.g., a force feedback joystick) to be documented within the same CAP_{SYS} record and can avoid some record duplication.

However, the algorithms to combine multiple CAPs do not exist at this time, since they are beyond the scope of this Thesis; thus, the second approach is inappropriate. Consequently, for the purposes of this Thesis, the first approach, where all parts of a system must be treated together as a single system, will be used. This approach has the advantage that it is similar to how the user examples above document each individual user at the CAP_{IC} level.

5.2.1 The CAP Approach to Systems

This Thesis considers a system to be a combination of hardware and software components that receive input from, and communicate output to, a human user (ISO, 1999a). The UARM model presented in Chapter 2 presents a system as a composite of processing capabilities, interaction capabilities, and data presented to a user through an interface. In terms of the CAP, this interface can be seen as the capability of a system to receive or transmit information in a specific modality.

For example, a microphone connected to a system is most useful if the system has software that uses it. In this example, "software" encompasses not just hardware drivers but, perhaps more importantly, applications that can use the input auditory modality data received by the microphone. Thus, the microphone is essentially useless, if there is no software that uses it.

This section presents three system examples expressing CAPs in terms of input (or output) modality capabilities.

The first example presents a CAP which may describe a typical computing system. It highlights the philosophy of ISO 9241-171 to ensure full use via the keyboard only. However, there is no restriction on the use of the mouse.

The second example builds onto the first by providing some additional software-based assistive technologies. It adds to the capability of full use via keyboard by also providing voice output.

The third example builds onto the previous two by providing additional software- and hardware-based assistive technologies. It shows how a system which cannot provide full accessible keyboard-based navigation can still provide some access to the keyboard and mouse as well as full voice input access.

As noted at the beginning of the chapter, linkages are described in terms of <*cap-name*, *linkage-type*> pairs and, where the *linkage-type* is not specified, the default value for the record type is to be assumed. In the case of systems, AND is the default *linkage-type*. The *linkage-type*s OR and XOR must be explicitly stated. Recall that AND implies that both operands of the conjunction "are included together" (not necessarily simultaneously) for access.

5.2.2 Standard System

5.2.2.1 Standard System Description

A desktop computer of the type normally found in many households. This computer has a microphone and speakers that could be used for alternative input/output modalities; however, only the already built-in AT consisting of the basic Computer Interface Access Enhancements developed by the Trace R&D Center of the University of Wisconsin-Madison[1] is available — no other third party provider AT is available.

The system has the following:

Input:

- Standard QWERTY keyboard and software that can read this input.
- Right-handed ergonomic two-button mouse with scroll wheel and software that can use this input.
- Microphone and software that can record or transmit via internet this input.

Output:

- 17 inch monitor capable of supporting all standard resolutions and software that supports this display.
- Stereo speakers and software that outputs various types of audio.

Thus, this system's CAP must at least document the following capabilities:

Auditory Input: The presence of a microphone suggests this system supports auditory input capabilities. Its software limits the range of these capabilities to general sounds to be recorded

[1] That is: StickyKeys, SlowKeys, BounceKeys, FilterKeys, MouseKeys, RepeatKeys, ToggleKeys, SoundSentry, ShowSounds, Time Out and SerialKeys (see ISO 9241-171).

and/or transmitted (e.g., over the internet). This means that the system has sound and music media auditory output capabilities, but cannot use this information as input data (e.g., speech recognition).

Auditory Output: The presence of stereo speakers suggests this system supports stereo auditory output capabilities. Its software supports audio playback meaning that the system is capable of playback of any previously recorded audio input. Its software also supports system generated audio output such as error alarms.

Visual Input: The absence of a camera or other visual input device suggests this system does not support visual input capabilities.

Visual Output: The presence of a monitor supporting all standard resolutions suggests this system supports visual output capabilities. Its software fully supports this display. This means that this system is capable of all forms of visual output via the monitor.

Tactile Input: The presence of a keyboard and mouse suggests this system supports some tactile input capabilities. Its software fully supports these input devices. This means that the system is capable of using text input via keyboard and generic mouse input for navigation and manipulation of user interfaces.

Tactile Output: The absence of an explicit tactile output device suggests this system does not support tactile output capabilities.

5.2.2.2 Standard System CAP

This system's CAP_{CF}s document the capability of using specific input and output modalities. We start with the CAP_{IC} for this system, shown in Table 5.33.

Table 5.33: A CAP for a "Standard System"

CAP_{IC} Identification	
Name	StandardSystem
Type	CAP_{SYS}
Qualifier	MAY
Description	This is the starting point of the system CAP for the Standard System example.
CAP_{IC} Linkages	
Higher-CAP_O	{<Root, >}
Peer-CAP_{IC}	
Lower-CAP_{IR}	{<System1-Tactile_Input, >, <System1-Auditory_Input, OR>}
Lower-CAP_{PF}	
Lower-CAP_{OT}	{<System1-Visual_Output, >, <System1-Auditory_Output, OR>}

Table 5.33 documents the start of this system's CAP:

- Higher-CAP$_O$ links to the Root CAP$_O$ specified at the beginning of the chapter.

In Table 5.33, the use of the *linkage-type* OR indicates that other modalities are available. For example, the system can support certain types of auditory input. Only those modalities that are ANDed (i.e., the default) indicate modalities that are required by the interface of the system to interact with it.

Now we document all of the input capabilities of this system. If we start with the tactile modality, we can document both the keyboard and mouse. The CAP$_{IR}$ for this modality is shown in Table 5.34.

Table 5.34: Tactile Input

Identification	
Type	CAP$_{IR}$
Name	System1-Tactile_Input
Qualifier	SHALL
Description	Tactile modality input capabilities. A standard PC US English QWERTY keyboard and a two-button mouse with a scroll wheel.
Linkages	
Higher-CAP$_{IC}$	{<StandardSystem, >}
Peer-CAP$_{IR}$	{<System1-Auditory_Input, >}
Peer-CAP$_{PF}$	
Peer-CAP$_{OT}$	{<System1-Visual_Output, OR>, <System1-Auditory_Output, OR>}
Lower-CAP$_M$	{<System1-Keyboard_Modality, >, <System1-LeftMouseButton, OR>, <System1-RightMouseButton, OR>, <System1-MouseWheel, OR>}
Lower-CAP$_C$	{<System1-Keyboard_Capability, >, <System1-Mouse_Capability, OR>}
Lower-CAP$_P$	
Communication capabilities	
Capacity	1
Parallel	1
Serial	1
Degree	
Intermittent	
Devoted	

Table 5.34 documents the system's capability to use tactile input:

- In this record, the use of the *linkage-type* OR with Lower-CAP$_M$ and Lower-CAP$_C$ indicates other available tactile modality interfaces and their capabilities, those that are ANDed (i.e., the default) indicate what is needed to use the system.

- As shown in the CAP$_{USE}$ examples, Peer-CAP$_{OT}$s are always ORed in CAP$_{IR}$s (and vice versa). This indicates only what peer CAP_{OT}s are available.

- Communication capabilities suggest that this IR can be used either in Parallel or Serial.

Drawing on the assumption discussed in Chapter 2, while in the real world this may not be the case, for the purposes of this Thesis, this Chapter will continue to follow the suggestion that system CFs should be defined as simple CFs capable of connecting to only a single channel. In the case of Table 5.34, this means that the system is only able to receive tactile input from either of the keyboard or mouse, but not from both.

Table 5.35 shows the CAP_M for the keys of the keyboard documented in Table 5.34.

Table 5.35: Keyboard Input Modality

Identification	
Type	CAP_M
Name	System1-Keyboard_Modality
Qualifier	SHALL
Description	The input keys of the keyboard.
Linkages	
Higher-CAP_{IR}	{<System1-Tactile_Input, >}
Higher-CAP_{OT}	
Higher-CAP_{PF}	
Peer-CAP_M	
Peer-CAP_C	{<System1-Keyboard_Capability, >}
Peer-CAP_P	
Modality-Specific Information	
ModalityType	TACTILE
MediaTypes	{`TextWritten`}
Language	{<`eng, Latn`>}

Table 5.35 documents this system's ability to receive text input via keyboard:

- *ModalityType* is TACTILE, because physical keyboards represent tactile modality input devices,

- *MediaTypes* indicates that written text is the type of media produced by keyboards, and

- *Language* is set to indicate that this keyboard is specific to Latin alphabets and the English language character set.

Keyboards can come in several sizes and layouts. For example, laptop keyboards are often reduced in size, BlackBerry™ keyboards are very small in size. Keyboard size is an important access issue for some users. For example, a user who has small hands might not be able to use a full-sized keyboard. These dimensions and ISO/IEC 9995-3 recognized national layout (i.e., QWERTY, the US English national layout) can be specified in a CAP_C as shown in Table 5.36 (International Organization for Standardization [ISO] and International Electrotechnical Commission [IEC], 2002).

Table 5.36: Keyboard Input Modality Capabilities

Identification	
Type	CAP$_{C}$
Name	System1-Keyboard_Capability
Qualifier	SHALL
Description	The keyboard's capabilities. Here the length, width, height, and layout of the keyboard are described.
Linkages	
Higher-CAP$_{IR}$	{<System1-Tactile_Input, >}
Higher-CAP$_{OT}$	
Higher-CAP$_{PF}$	
Peer-CAP$_{M}$	{<System1-Keyboard_Modality, >}
Peer-CAP$_{C}$	
Peer-CAP$_{P}$	
Capability-Specific Information	
Capability name	Length
Capability instance	0
Capability values	{<45.72, 45.72, cm>}
Capability name	Width
Capability instance	0
Capability values	{<20.32, 20.32, cm>}
Capability name	Height
Capability instance	0
Capability values	{<5.715, 5.715, cm>}
Capability name	other-ISO/IEC9995-3-KeyboardLayout
Capability instance	0
Capability values	en_US

We document the mouse starting in Table 5.37.

Table 5.37: Left Mouse Button Input Modality

Identification	
Type	CAP$_{M}$
Name	System1-LeftMouseButton
Qualifier	MAY
Description	The modality of the left-side mouse button.
Linkages	
Higher-CAP$_{IR}$	{<System1-Tactile_Input, >}
Higher-CAP$_{OT}$	
Higher-CAP$_{PF}$	
Peer-CAP$_{M}$	{<System1-RightMouseButton, OR>, <System1-MouseWheel, OR>}
Peer-CAP$_{C}$	{<Mouse_Capability, >}
Peer-CAP$_{P}$	
Modality-Specific Information	
ModalityType	TACTILE
MediaTypes	{`UNKNOWN`}
Language	{`<NONE, NIL>`}

According to the record in Table 5.34, the mouse's buttons and scroll wheel are treated in separate CAP_Ms, because they are each operable controls with their own modalities and potential capabilities. Thus, Table 5.37 shows the CAP for the left mouse button.

Table 5.37 documents that this mouse button can receive tactile input:

- *Qualifier* is MAY to indicate the optional use of the mouse.
- The *linkage-type* OR is specified on Peer-CAP_Ms at this level to indicate other mouse buttons could also be used,
- The *linkage-type* AND is implied on Peer-CAP_Cs at this level to indicate the capabilities associated with this CAP_M,
- *ModalityType* is set to TACTILE because mice represent a tactile input method,
- *MediaTypes* is set to UNKNOWN because the mouse does not support any of the explicit CAP media types listed in Table 3.9, and
- *Language* is set to indicate that mice are language neutral.

Since the right-hand mouse button and the scroll wheel have the same settings, their records will not be shown here.

If a left-handed user uses an ergonomically right-handed-specific mouse, they may experience discomfort. ISO/IEC 24751 allows us to specify whether an input device is explicitly ergonomically configured for a right- or left-handed person.

Mice come in several sizes. For example, mini (extra small) mice are commonly used with laptops. Mouse size can impact one user across different ages, as well as different users with different hand sizes. Although information on hand size was not provided in any of the user examples above, mouse size can be documented in a system's CAP to enable users, who do have hand size limitations, to choose mice within a specific size.

Since there is no current ISO standard that provides an explicit definition of pointing device sizes, this information must be encoded using SI units. The actual physical size of the mouse is not easily comparable to users; thus, to make this specification even more useful when comparing users, the handsize the mouse is designed for is specified instead. Handsize is measured from the crease of the wrist closest to the hand out to the end of the middle finger (for more information see `http://www.contourdesign.com/perfit_size.htm`).

Thus, Table 5.38 shows what the CAP_C for this mouse might look like. Note that, in this chapter, all examples using ISO/IEC 24751, in the interests of space and readability, only show the portion of the XML code needed.

Table 5.38: An Ergonomically Right-Handed Mouse for a Medium-Sized Hand

Identification	
Type	CAP$_C$
Name	Mouse_Capability
Qualifier	MAY
Description	Additional properties and capabilities of the mouse. Here the size and handedness of the mouse is described.
Linkages	
Higher-CAP$_{IR}$	{<System1-Tactile_Input, >}
Higher-CAP$_{OT}$	
Higher-CAP$_{PF}$	
Peer-CAP$_M$	{<System1-LeftMouseButton, >, <System1-RightMouseButton, OR>, <System1-MouseWheel, OR>}
Peer-CAP$_C$	
Peer-CAP$_P$	
Capability-Specific Information	
Capability name	Handsize
Capability instance	0
Capability values	{<17, 19, cm>}
Capability name	other-ISO/IEC24751-handedness
Capability instance	0

Capability values	`...`
	`<alternativePointing>`
	`   <handedness value=‘‘right’’>`
	`</alternativePointing>`
	`...`

Table 5.38 documents this mouse's capabilities:

- The *linkage-type* OR may or may not be specified on Peer-CAP$_M$s at this level. If they are ANDed to this CAP$_C$, it shows that they go together. If they are ORed, it indicates alternatives.

- The mouse Capability handsize indicates this mouse is considered "medium-sized", and

- The handedness Capability documents that this mouse is ergonomically right-handed.

Note that this CAP does not force the user to use all of the mouse buttons. It only documents the presence of the mouse and the number of buttons available. The right-most mouse button has been ORed to show that back-clicking (in this case, via right-click) is available but not the primary mode of interaction with the mouse (i.e., the left-most button). This is important because of the access issues (e.g., due to repetitive strain injury, sprained wrist, etc.) which can be associated with using a mouse (e.g., Fagarasanu & Kumar, 2003).

The only other input modality supported by this system is the auditory modality via the microphone and its supporting software. The CAP$_{IR}$ is shown in Table 5.39.

Table 5.39: Auditory Input Capabilities

Identification	
Type	CAP$_{IR}$
Name	System1-Auditory_Input
Qualifier	MAY
Description	The system's auditory input capabilities using a PC microphone and its supporting software.
Linkages	
Higher-CAP$_{IC}$	{<StandardSystem, OR>}
Peer-CAP$_{IR}$	{<System1-Tactile_Input, >}
Peer-CAP$_{PF}$	
Peer-CAP$_{OT}$	{<System1-Visual_Output, OR>, <System1-Auditory_Output, OR>}
Lower-CAP$_{M}$	{<System1-Microphone_Modality, >}
Lower-CAP$_{C}$	{<System1-Microphone_Capability, >}
Lower-CAP$_{P}$	
Communication capabilities	
Capacity	1
Parallel	1
Serial	1
Degree	
Intermittent	
Devoted	

A microphone supports auditory modality input as shown in Table 5.40.

Table 5.40: Auditory Input Modality

Identification	
Type	CAP$_{M}$
Name	System1-Microphone_Modality
Qualifier	MAY
Description	The supported modalities of the microphone and associated software.
Linkages	
Higher-CAP$_{IR}$	{<System1-Auditory_Input, >}
Higher-CAP$_{OT}$	
Higher-CAP$_{PF}$	
Peer-CAP$_{M}$	
Peer-CAP$_{C}$	{<System1-Microphone_Capability, >}
Peer-CAP$_{P}$	
Modality-Specific Information	
ModalityType	AUDITORY
MediaTypes	{`Sound`, `Music`}
Language	{<`NONE`, `NIL`>}

Table 5.40 documents this system's ability to receive auditory data:

- *MediaTypes* indicates that a microphone can be used to input music or any other sounds, and
- *Language* is set to indicate that microphones are language neutral.

In addition to various other possibilities, all microphones have one major capability: their supported input frequency response. For a microphone, frequency response is a, "measure of the consistency with which it translates a given sound pressure level into an audio signal level at different frequencies" (Davis & Jones, 1989). Ideally, a microphone translates an SPL to the same audio signal level (i.e., a "flat" response). This capability is important, because it can impact system software dependent on microphone input (e.g., recording software, dictation software, etc.). The CAP_C in Table 5.41 is used to document this capability. To ensure realistic specifications of the hardware capabilities, the CAP shown in Table 5.41 is based on the specifications of the Altec Lansing ABM200 (for details see: http://www.alteclansing.com/product_details.asp?pID=ABM200).

Table 5.41: Auditory Modality Input Capabilities

Identification	
Type	CAP_C
Name	System1-Microphone_Capability
Qualifier	MAY
Description	Additional properties and capabilities of the microphone. Here the input frequency of the microphone is described.
Linkages	
Higher-CAP_{IR}	{<System1-Auditory_Input, >}
Higher-CAP_{OT}	
Higher-CAP_{PF}	
Peer-CAP_M	{<System1-Microphone_Modality, >}
Peer-CAP_C	
Peer-CAP_P	
Capability-Specific Information	
Capability name	FrequencyResponse
Capability instance	0
Capability values	{<50, 16000, Hz>}

This ends the discussion of this system's input capabilities. Now we document all of the output capabilities of this system: visual and auditory. Table 5.42 shows the CAP_{OT} for the visual output modality.

Table 5.42: Visual Output

Identification	
Type	CAP$_{OT}$
Name	System1-Visual_Output
Qualifier	SHALL
Description	A 17 inch monitor (capable of supporting all standard resolutions) and its supporting software.
Linkages	
Higher-CAP$_{IC}$	{<StandardSystem, >}
Peer-CAP$_{IR}$	{<System1-Tactile_Input, OR>, <System1-Auditory_Input, OR>}
Peer-CAP$_{PF}$	
Peer-CAP$_{OT}$	{<System1-Auditory_Output, >}
Lower-CAP$_{M}$	{<System1-Monitor_Modality, >}
Lower-CAP$_{C}$	{<System1-Screen_Capability, >}
Lower-CAP$_{P}$	
Communication capabilities	
Capacity	1
Parallel	1
Serial	1
Degree	
Intermittent	
Devoted	

Table 5.42 documents the system's capability to produce visual output:

- As shown in the CAP$_{USE}$ examples, Peer-CAP$_{IR}$s are always ORed in CAP$_{OT}$s (and vice versa). This indicates only what peer CAP$_{IR}$s are available.

Monitors can support all media types in the visual modality and are, in themselves, language neutral. This is documented with the CAP$_{M}$ in Table 5.43.

Table 5.43: Visual Modality Output

Identification	
Type	CAP$_M$
Name	System1-Monitor_Modality
Qualifier	SHALL
Description	The output modality of the monitor.
Linkages	
Higher-CAP$_{IR}$	
Higher-CAP$_{OT}$	{<System1-Visual_Output, >}
Higher-CAP$_{PF}$	
Peer-CAP$_M$	
Peer-CAP$_C$	{<System1-Screen_Capability, >}
Peer-CAP$_P$	
Modality-Specific Information	
ModalityType	VISUAL
MediaTypes	{`TextWritten`, `TextSigned`, `Picture`, `VisualModel`, `DynamicVisualModel`, `Movie`}
Language	{<`NONE`, `NIL`>}

Monitors can be further described in terms of the size of their screen and their supported colours and resolutions. This is done in a CAP$_C$ as shown in Table 5.44.

Table 5.44: Visual Modality Output Capabilities

Identification	
Type	CAP$_C$
Name	System1-Screen_Capability
Qualifier	SHALL
Description	The screen size, supported colours, and supported resolutions of the screen.
Linkages	
Higher-CAP$_{IR}$	
Higher-CAP$_{OT}$	{<System1-Visual_Output, >}
Higher-CAP$_{PF}$	
Peer-CAP$_M$	{<System1-Monitor_Modality, >}
Peer-CAP$_C$	
Peer-CAP$_P$	
Capability-Specific Information	
Capability name	Colours
Capability instance	0
Capability values	{<810,1620, THz>}
Capability name	Resolution
Capability instance	0
Capability values	{<1024, 768, pixels>}
Capability name	Screensize
Capability instance	0
Capability values	{<43.18, 43.18, cm>}

Monitors support several different resolutions, so only the current resolution is documented in the above Table 5.44. Screen size can be considered in two senses:

- the length and width of the viewable area, or
- the diagonal length of the screen.

The first approach may be more meaningful and may mean using more than one Capability name. The second approach is based on current consumer convention. The second approach has been used in this example. Also, this monitor has a 17 inch (43.18 cm) screen size. This fully describes the visual modality output capabilities.

The CAP for the auditory modality output capabilities begins in Table 5.45.

Table 5.45: Auditory Output

Identification	
Type	CAP_{OT}
Name	System1-Auditory_Output
Qualifier	MAY
Description	The system's auditory output capabilities, based on a pair of speakers capable of mono and stereo output and their associated software.
Linkages	
Higher-CAP_{IC}	{<StandardSystem, OR>}
Peer-CAP_{IR}	{<System1-Tactile_Input, OR>, <System1-Auditory_Input, OR>}
Peer-CAP_{PF}	
Peer-CAP_{OT}	{<System1-Visual_Output, >}
Lower-CAP_{M}	{<System1-LeftSpeaker_Modality, OR>, <System1-RightSpeaker_Modality, OR>}
Lower-CAP_{C}	{<System1-LeftSpeaker_Capability, OR>, <System1-RightSpeaker_Capability, OR>}
Lower-CAP_{P}	
Communication capabilities	
Capacity	1
Parallel	1
Serial	1
Degree	
Intermittent	
Devoted	

Each speaker's modalities and capabilities are documented in separate records. This allows exclusion or inclusion of each speaker as needed (e.g., to document stereo capability). Speakers support the auditory modality and its media types but are, in themselves, language neutral as the record in Table 5.46 states.

Table 5.46: Auditory Output Modality

Identification	
Type	CAP$_{M}$
Name	System1-LeftSpeaker_Modality
Qualifier	MAY
Description	The modality of the left-hand speaker and its supporting software.
Linkages	
Higher-CAP$_{IR}$	
Higher-CAP$_{OT}$	{<System1-Auditory_Output, OR>}
Higher-CAP$_{PF}$	
Peer-CAP$_{M}$	{<System1-RightSpeaker_Modality, OR>}
Peer-CAP$_{C}$	{<System1-LeftSpeaker_Capability, >, <System1-RightSpeaker_Capability, OR>}
Peer-CAP$_{P}$	
Modality-Specific Information	
ModalityType	AUDITORY
MediaTypes	{`Sound`, `Music`}
Language	{<`NONE`, `NIL`>}

The CAP$_{M}$ of the right-hand speaker is similar and thus not shown here. Note that in Table 5.46, *System1-LeftSpeaker_Capability* is ANDed to this record, but *System1-RightSpeaker_Modality* and *System1-LeftSpeaker_Capability* are ORed to this record. The *System1-LeftSpeaker_Capability* record is intended to go with the record in Table 5.46, but other records can be used instead. The CAP$_{C}$ shown in Table 5.47 also uses this approach.

In addition to various other possibilities, all speakers have two major capabilities: their frequency response, and their peak intensity. For a speaker, frequency response is an indication of its accuracy for reproducing audio. For a speaker, peak intensity is an indication of how loudly it can reproduce audio[2]. These capabilities are important because they impact the quality of the sound output to the user. The CAP$_{C}$ for the left speaker is shown in Table 5.47. This table is based on the specifications of the Altec Lansing VS2320 (for details see: `http://www.alteclansing.com/product_details.asp?pID=VS2320`).

[2]See `http://www.myhometheater.homestead.com/splcalculator.html` to learn how to calculate peak intensity.

Table 5.47: Auditory Output Capabilities

Identification	
Type	CAP$_C$
Name	System1-LeftSpeaker_Capability
Qualifier	MAY
Description	Additional properties and capabilities of the left-hand speaker. Here the frequency response and peak intensity are described.
Linkages	
Higher-CAP$_{IR}$	
Higher-CAP$_{OT}$	{<System1-Auditory_Output, >}
Higher-CAP$_{PF}$	
Peer-CAP$_M$	{<System1-LeftSpeaker_Modality, >, <System1-RightSpeaker_Modality, OR>}
Peer-CAP$_C$	{<System1-RightSpeaker_Capability, OR>}
Peer-CAP$_P$	
Capability-Specific Information	
Capability name	FrequencyResponse
Capability instance	0
Capability values	{<85, 18000, Hz>}
Capability name	PeakIntensity
Capability instance	0
Capability values	{<0, 106.3, dB>}

The CAP$_C$ of the right-hand speaker is similar and thus not shown here. This ends the description of this system's output capabilities.

At a high level, Table 5.48 describes the structure of this system's CAP.

Table 5.48: Structure of CAP_{SYS} for "Standard System"

Name	Type	Unary Operator	Tables
1.0 StandardSystem	CAPSYS	MAY	Table 5.33
1.1 System1-Tactile_Input	CAPIR	SHALL	Table 5.34
1.1.1 System1-Keyboard_Modality	CAPM	SHALL	Table 5.35
1.1.2 System1-Keyboard_Capability	CAPC	SHALL	Table 5.36
1.1.3 System1-LeftMouseButton	CAPM	MAY	Table 5.37
1.1.4 System1-RightMouseButton	CAPM	MAY	Like Table 5.37
1.1.5 System1-MouseWheel	CAPM	MAY	Like Table 5.37
1.1.6 System1-Mouse_Capability	CAPC	MAY	Table 5.38
1.2 System1-Auditory_Input	CAPIR	MAY	Table 5.39
1.2.1 System1-Microphone_Modality	CAPM	MAY	Table 5.40
1.2.2 System1-Microphone_Capability	CAPC	MAY	Table 5.41
1.3 System1-Visual_Output	CAPOT	SHALL	Table 5.42
1.3.1 System1-Monitor_Modality	CAPM	SHALL	Table 5.43
1.3.2 System1-Screen_Capability	CAPC	SHALL	Table 5.44
1.4 System1-Auditory_Output	CAPOT	MAY	Table 5.45
1.4.1 System1-LeftSpeaker_Modality	CAPM	MAY	Table 5.46
1.4.2 System1-RightSpeaker_Modality	CAPM	MAY	Like Table 5.46
1.4.3 System1-LeftSpeaker_Capability	CAPC	MAY	Table 5.47
1.4.4 System1-RightSpeaker_Capability	CAPC	MAY	Like Table 5.47

5.2.3 System with Some Additional Software AT Support

5.2.3.1 System with Software AT Support Description

This System is a desktop computer like the Standard System described above. Its operating system provides software AT beyond the basic Computer Interface Access Enhancements. Specifically, the system comes with a screen reader for text-to-speech conversion and screen magnification built-in. Instead of a microphone and speakers, this computer has a headset with boom microphone that could be used for alternative input/output modalities.

The system can be described in terms of the following:

Input hardware:

- Standard QWERTY keyboard and software that can read this input.
- Right-handed ergonomic two-button mouse with a scroll wheel and software that can use this input.
- Boom microphone and software that can record or transmit via internet this input.

Output hardware:

- 15 inch monitor capable of supporting all standard resolutions, but set to 800x600 resolution and software that supports this display.

- Stereo headset and software that outputs various types of audio including text-to-speech output.

Thus, this system's CAP must at least document the following capabilities:

Auditory Input: The presence of a boom microphone suggests this system supports auditory input capabilities. Its software limits the range of these capabilities to general sounds to be recorded and/or transmitted (e.g., over the internet). This means that the system has sound and music media auditory output capabilities, but cannot use this information as input data (e.g., speech recognition).

Auditory Output: The presence of a stereo headset suggests this system supports stereo auditory output capabilities. Its software supports audio playback meaning that the system is capable of playback of any previously recorded audio input. Its software also supports system-generated audio output, such as error alarms and speech output generated by this system's text-to-speech conversion software.

Visual Input: The absence of a camera or other visual input device suggests this system does not support visual input capabilities.

Visual Output: The presence of a monitor supporting all standard resolutions suggests this system supports visual output capabilities. Its software fully supports this display. This means that this system is capable of all forms of visual output via the monitor.

Tactile Input: The presence of a keyboard and mouse suggests this system supports some tactile input capabilities. Its software fully supports these input devices. This means that the system is capable of using text input via keyboard and generic mouse input for navigation and manipulation of user interfaces.

Tactile Output: The absence of an explicit tactile output device suggests this system does not support tactile output capabilities.

Processing Capabilities: The presence of a screen reader (text-to-speech) and screen magnification software suggests that this system supports some output processing capabilities.

5.2.3.2 System with Software AT Support CAP

This system's CAP_{CF}s document the capability of using specific input and output modalities. We start with the CAP_{IC} for this system, shown in Table 5.49.

Table 5.49: A CAP for "System with Some Additional Software AT Support"

CAP_{IC} Identification	
Name	AnotherStandardSystem
Type	CAP_{SYS}
Qualifier	MAY
Description	This is the starting point of the system CAP for the System with some additional software AT support example.
CAP_{IC} Linkages	
Higher-CAP_O	{<Root, >}
Peer-CAP_{IC}	
Lower-CAP_{IR}	{<System2-Tactile_Input, >, <System2-Auditory_Input, OR>}
Lower-CAP_{PF}	{<System2-Screenreader, OR>, <System2-Magnification, OR>}
Lower-CAP_{OT}	{<System2-Visual_Output, >, <System2-Auditory_Output, OR>}

Table 5.49 documents the start of this system's CAP:

- Lower-CAP_{PF} links to CAP_{PF}s describing the screen reader and screen magnifier software capabilities.

Table 5.50: Auditory Input

Identification	
Type	CAP_{IR}
Name	System2-Auditory_Input
Qualifier	MAY
Description	The system's auditory input capabilities using a headset boom microphone and its supporting software.
Linkages	
Higher-CAP_{IC}	{<AnotherStandardSystem, >}
Peer-CAP_{IR}	{<System2-Tactile_Input, >}
Peer-CAP_{PF}	{<System2-Screenreader, >, <System2-Magnification, >}
Peer-CAP_{OT}	{<System2-Visual_Output, OR>, <System2-Auditory_Output, OR>}
Lower-CAP_M	{<System2-BoomMic_Modality, >}
Lower-CAP_C	{<System2-BoomMic_Capability, >}
Lower-CAP_P	
Communication capabilities	
Capacity	1
Parallel	1
Serial	1
Degree	
Intermittent	
Devoted	

Now we document all of the input capabilities of this system. If we start with the tactile modality, we can document both the keyboard and mouse. Since this is treated in the same manner as the Standard System example above, we will not show it here.

The only other input modality supported by this system is the auditory modality via the boom microphone of the headset and its supporting software. The CAP$_{IR}$ is shown in Table 5.50 above.

As with the microphone described earlier, a boom microphone supports auditory modality input and is language neutral. This is documented in Table 5.51.

Table 5.51: Auditory Modality Input

Identification	
Type	CAP$_M$
Name	System2-BoomMic_Modality
Qualifier	MAY
Description	The supported modalities of the boom microphone and associated software.
Linkages	
Higher-CAP$_{IR}$	{<System2-Auditory_Input, >}
Higher-CAP$_{OT}$	
Higher-CAP$_{PF}$	
Peer-CAP$_M$	
Peer-CAP$_C$	{<System2-BoomMic_Capability, >}
Peer-CAP$_P$	
Modality-Specific Information	
ModalityType	AUDITORY
MediaTypes	{`Sound`, `Music`}
Language	{<`NONE, NIL`>}

As mentioned above, in addition to various other possibilities, all microphones have one major capability: their supported input frequency response. This capability is important because it can impact system software dependent on boom microphone input (e.g., recording software, dictation software). Table 5.52 shows the boom microphone's CAP$_C$ and is based on the specifications of the Altec Lansing AHS423 (for details see: `http://www.alteclansing.com/product_details.asp?pID=AHS423)`

Table 5.52: Auditory Input Modality Capabilities

Identification	
Type	CAP_C
Name	System2-BoomMic_Capability
Qualifier	MAY
Description	Additional properties and capabilities of the boom microphone (i.e., input frequency).
Linkages	
Higher-CAP_{IR}	{<System2-Auditory_Input, >}
Higher-CAP_{OT}	
Higher-CAP_{PF}	
Peer-CAP_M	{<System2-BoomMic_Modality, >}
Peer-CAP_C	
Peer-CAP_P	
Capability-Specific Information	
Capability name	InputFrequency
Capability instance	0
Capability values	{<100, 10000, Hz>}

This ends the discussion of this system's input capabilities. Now we document all of the output capabilities of this system: the monitor and earphones of the headset.

Since, with the exception of specific capabilities, the monitor is treated in the same manner as the Standard System example above, we will only show its CAP_C in Table 5.53.

Table 5.53: Visual Output Modality Capabilities

Identification	
Type	CAP_C
Name	System2-Screen_Capability
Qualifier	SHALL
Description	The screen size, supported colours, and supported resolutions of the screen.
Linkages	
Higher-CAP_{IR}	
Higher-CAP_{OT}	{<System2-Visual_Output, >}
Higher-CAP_{PF}	
Peer-CAP_M	{<System2-Monitor_Modality, >}
Peer-CAP_C	
Peer-CAP_P	
Capability-Specific Information	
Capability name	Colours
Capability instance	0
Capability values	{<810,1620, THz>}
Capability name	Resolution
Capability instance	0
Capability values	{<800, 600, pixels>}
Capability name	Screensize
Capability instance	0
Capability values	{<38.1, 38.1, cm>}

Within their CAP$_C$s, monitors can be described in terms of the size of their screen and their supported colours and resolutions. Monitors support several different resolutions, so only the current resolution is documented here. In this case, the monitor is currently set to an 800x600 resolution. Screen size can be considered in two senses:

- the length and width of the viewable area, or
- the diagonal length of the screen.

The first approach may be more meaningful and may mean using more than one Capability name. The second approach is based on current consumer convention. The second approach has been used in this example. The monitor described in Table 5.53 has a 15 inch (38.1 cm) screen size.

The CAP for the auditory modality output capabilities of this system begins in Table 5.54.

Table 5.54: Earphone-Based Auditory Output

Identification	
Type	CAP$_{OT}$
Name	System2-Auditory_Output
Qualifier	MAY
Description	The system's auditory output capabilities based on a pair of earphones capable of mono and stereo output and their associated software.
Linkages	
Higher-CAP$_{IC}$	{<AnotherStandardSystem, >}
Peer-CAP$_{IR}$	{<System2-Tactile_Input, OR>, <System2-Auditory_Output, OR>}
Peer-CAP$_{PF}$	{<System2-Screenreader, OR>, <System2-Magnification, OR>}
Peer-CAP$_{OT}$	{<System2-Visual_Output, >}
Lower-CAP$_M$	{<System2-LeftEarphone_Modality, OR>, <System2-RightEarphone_Modality, OR>}
Lower-CAP$_C$	{<System2-LeftEarphone_Capability, OR>, <System2-RightEarphone_Capability, OR>}
Lower-CAP$_P$	
Communication capabilities	
Capacity	1
Parallel	1
Serial	1
Degree	
Intermittent	
Devoted	

Each earphone's modalities and capabilities are documented in separate records. This allows exclusion or inclusion of each earphone as needed (e.g., to document stereo capability). Earphones support the auditory modality and its associated media types but are, in themselves, language neutral. Thus, Table 5.55 shows what the CAP$_M$ for the left-hand earphone might look like.

Table 5.55: Earphone-Based Auditory Output Modality

Identification	
Type	CAP_M
Name	System2-LeftEarphone_Modality
Qualifier	MAY
Description	The modality of the left-hand earphone and its supporting software.
Linkages	
Higher-CAP_{IR}	
Higher-CAP_{OT}	{<System2-Auditory_Output, OR>}
Higher-CAP_{PF}	
Peer-CAP_M	{<System2-RightEarphone_Modality, OR>}
Peer-CAP_C	{<System2-LeftEarphone_Capability, >, <System2-RightEarphone_Capability, OR>}
Peer-CAP_P	
Modality-Specific Information	
ModalityType	AUDITORY
MediaTypes	{`TextSpoken,` `Sound,` `Music`}
Language	{<`NONE, NIL`>}

The CAP_M of the right-hand earphone is similar and thus not shown here. As mentioned above, in addition to various other possibilities, all speakers, including earphones, have two major capabilities: their frequency response, and their peak intensity[3]. These capabilities are important because they impact the quality of the sound output to the user. The CAP_C for the left earphone is shown in Table 5.56. This table is again based on the specifications of the Altec Lansing AHS423. Note that the peak intensity of this earphone was calculated at 90dB sensitivity, 0.1 W amplifier power, and a distance of 1.3 cm (one-half inch) from the earphone.

[3]For information on how to calculate peak intensity see `http://www.myhometheater.homestead.com/splcalculator.html`.

Table 5.56: Auditory Output Modality Capabilities

Identification	
Type	CAP_C
Name	System2-LeftEarphone_Capability
Qualifier	MAY
Description	Additional properties and capabilities of the left-hand speaker. Here the frequency response and peak intensity are described.
Linkages	
Higher-CAP_{IR}	
Higher-CAP_{OT}	{<System2-Auditory_Output, OR>}
Higher-CAP_{PF}	
Peer-CAP_M	{<System2-LeftEarphone_Modality, >, <System2-RightEarphone_Modality, OR>}
Peer-CAP_C	{<System2-RightEarphone_Capability, OR>}
Peer-CAP_P	
Capability-Specific Information	
Capability name	FrequencyResponse
Capability instance	0
Capability values	{<20, 20000, Hz>}
Capability name	PeakIntensity
Capability instance	0
Capability values	{<0, 121, dB>}

The CAP of the right-hand earphone is similar and thus not shown here. This ends the description of this system's output capabilities.

Since the system also comes with screen reading and screen magnification software, we need to document the CAP_{PF}s of these software features. First, we start with the screen reader CAP_{PF} record as shown in Table 5.57.

Table 5.57: A Screen Reader CAP

Identification	
Type	CAP$_{PF}$
Name	System2-Screenreader
Qualifier	MAY
Description	Screen reader software for text-to-speech output.
Linkages	
Higher-CAP$_{IC}$	{<AnotherStandardSystem, >}
Peer-CAP$_{IR}$	{<System2-Tactile_Input, OR>, <System2-Auditory_Output, OR>}
Peer-CAP$_{PF}$	{<System2-Magnification, OR>}
Peer-CAP$_{OT}$	{<System2-Visual_Output, OR>, <System2-Auditory_Output, >}
Lower-CAP$_{M}$	{<System2-Screenreader_Input, >, <System2-Screenreader_Output, >}
Lower-CAP$_{C}$	{<System2-Screenreader_Capability, >}
Lower-CAP$_{P}$	{<System2-Screenreader_Processing, >}
Communication capabilities	
Capacity	1
Parallel	1
Serial	1
Degree	
Intermittent	
Devoted	

Table 5.57 documents that this system has the ability to convert text output to speech:

- Lower-CAP$_{M}$ links the screen reader's two CAP$_{M}$s, one for text input and the other for speech output, and
- Lower-CAP$_{P}$ links to the screen reader's CAP$_{P}$, which documents whether the screen reader is a pass through AT, and whether it transforms its input content stream.

Note that Table 5.57 shows an AND relationship with the Peer-CAP$_{OT}$ "System2-Auditory_Output". This relationship is necessary because text-to-speech screen reading software is dependent on the system's auditory output capabilities.

This software works primarily with text input (i.e., text-to-speech), but also can work with limited graphic inputs (e.g., reading screen elements via optical character recognition). Thus, the CAP$_{M}$ in Table 5.58 describes the screen reader's supported inputs.

Table 5.58: Screen Reader Input Modalities

Identification	
Type	CAP_M
Name	System2-Screenreader_Input
Qualifier	MAY
Description	The modality of the screen reader inputs.
Linkages	
Higher-CAP_{IR}	
Higher-CAP_{OT}	
Higher-CAP_{PF}	{<System2-Screenreader, >}
Peer-CAP_M	{<System2-Screenreader_Output, >}
Peer-CAP_C	{<System2-Screenreader_Capability, >}
Peer-CAP_P	{<System2-Screenreader_Processing, >}
Modality-Specific Information	
ModalityType	VISUAL
MediaTypes	{TextWritten, Picture}
Language	{<eng, Latn>}

In addition, since this screen reader provides speech output, the CAP_M in Table 5.59 describes its supported outputs.

Table 5.59: Screen Reader Output Modalities

Identification	
Type	CAP_M
Name	System2-Screenreader_Output
Qualifier	MAY
Description	The modality of the screen reader outputs.
Linkages	
Higher-CAP_{IR}	
Higher-CAP_{OT}	
Higher-CAP_{PF}	{<System2-Screenreader, >}
Peer-CAP_M	{<System2-Screenreader_Input, >}
Peer-CAP_C	{<System2-Screenreader_Capability, >}
Peer-CAP_P	{<System2-Screenreader_Processing, >}
Modality-Specific Information	
ModalityType	AUDITORY
MediaTypes	{TextSpoken}
Language	{<eng, Latn>}

If desired, a CAP_C can be added to state-specific default settings for the screen reader using ISO/IEC 24751 tags. This is shown in Table 5.60.

Table 5.60: Screen Reader Output Capabilities

Identification	
Type	CAP_C
Name	System2-Screenreader_Capability
Qualifier	MAY
Description	The capabilities of the screen reader outputs.
Linkages	
Higher-CAP_{IR}	
Higher-CAP_{OT}	
Higher-CAP_{PF}	{<System2-Screenreader, >}
Peer-CAP_M	{<System2-Screenreader_Output, >}
Peer-CAP_C	
Peer-CAP_P	{<System2-Screenreader_Processing, >}
Capability-Specific Information	
Capability name	other-ISO/IEC24751-screenreader
Capability instance	0
Capability values	... <screenReader> <screenReaderGeneric> <speechRate usage=‘‘preferred’’ value=‘‘180’’> <pitch usage=‘‘optionallyUse’’ value=‘‘0.5’’> </screenReaderGeneric> </screenReader> ...

In addition, CAP_Ps can also contain capability information as shown in Table 5.61.

Table 5.61: Screen Reader Capabilities

Identification	
Type	CAP_P
Name	System2-Screenreader_Processing
Qualifier	MAY
Description	Additional properties and capabilities of the screen reader. Here the processing function is described.
Linkages	
Higher-CAP_{IR}	
Higher-CAP_{OT}	
Higher-CAP_{PF}	{<System2-Screenreader, >}
Peer-CAP_M	{<System2-Screenreader_Input, >, <System2-Screenreader_Output, >}
Peer-CAP_C	{<System2-Screenreader_Capability, >}
Peer-CAP_P	
Processing-Specific Information	
Capabilities	
Capability name	other-ISO/IEC24751-screenreader
Capability instance	1

Capability values	`...`
	`<screenReader>`
	`   <screenReaderGeneric>`
	`      <link`
	`         usage=‘‘required’’`
	`         value=‘‘speakLink’’>`
	`      <volume`
	`         usage=‘‘required’’`
	`         value=‘‘0.5’’>`
	`   </screenReaderGeneric>`
	`</screenReader>`
	`...`

Connectivity	
PassThrough	YES
Transformed	YES
Transformations	
MediaTransformation	YES
LangTransformation	NO

The CAP_P in Table 5.61 states that all content passes through the screen reader unmodified (i.e., the visual input content stream is not changed or closed off) as well as transformed into speech (i.e., media transformation).

The CAP_{PF} for screen magnification software is handled similarly to that for screen reading software, except that the input and output modality types are the same (i.e., visual). The media is not transformed (i.e., it is always visual output). Thus, the CAP_P for screen magnification appears in Table 5.62.

Table 5.62: A Screen Magnifier CAP

Identification	
Type	CAP_P
Name	System2-Magnification_Processing
Qualifier	MAY
Description	Additional properties and capabilities of the screen magnifier. Here the processing function is described.
Linkages	
Higher-CAP_{IR}	
Higher-CAP_{OT}	
Higher-CAP_{PF}	{<System2-Magnification, >}
Peer-CAP_M	{<System2-Magnification_Modality, >}
Peer-CAP_C	{<System2-Magnification_Capability, >}
Peer-CAP_P	
Processing-Specific Information	
Capabilities	
Capability name	
Capability instance	
Capability values	
Connectivity	
PassThrough	YES
Transformed	YES
Transformations	
MediaTransformation	NO
LangTransformation	NO

This ends the description of this system's processing capabilities.

At a high level, Table 5.63 describes the structure of this system's CAP.

Table 5.63: Structure of CAP_{SYS} for "System with Some Additional Software AT Support"

Name	Type	Unary Operator	Tables
1.0 AnotherStandardSystem	CAPSYS	MAY	Table 5.49
1.1 System2-Tactile_Input	CAPIR	SHALL	Like Table 5.33
1.1.1 System2-Keyboard_Modality	CAPM	SHALL	Like Table 5.34
1.1.2 System2-Keyboard_Capability	CAPC	SHALL	Like Table 5.35
1.1.3 System2-LeftMouseButton	CAPM	MAY	Like Table 5.37
1.1.4 System2-RightMouseButton	CAPM	MAY	Like Table 5.37
1.1.5 System2-MouseWheel	CAPM	MAY	Like Table 5.37
1.1.6 System2-Mouse_Capability	CAPC	MAY	Like Table 5.38
1.2 System2-Auditory_Input	CAPIR	MAY	Table 5.50
1.2.1 System2-BoomMic_Modality	CAPM	MAY	Table 5.51
1.2.2 System2-BoomMic_Capability	CAPC	MAY	Table 5.52
1.3 System2-Visual_Output	CAPOT	SHALL	Like Table 5.42
1.3.1 System2-Monitor_Modality	CAPM	SHALL	Like Table 5.43
1.3.2 System2-Screen_Capability	CAPC	SHALL	Table 5.53
1.4 System2-Auditory_Output	CAPOT	MAY	Table 5.54
1.4.1 System2-LeftEarphone_Modality	CAPM	MAY	Table 5.55
1.4.2 System2-RightEarphone_Modality	CAPM	MAY	Like Table 5.55
1.4.3 System2-LeftEarphone_Capability	CAPC	MAY	Table 5.56
1.4.4 System2-RightEarphone_Capability	CAPC	MAY	Like Table 5.56
1.5 System2-Screenreader	CAPPF	MAY	Table 5.57
1.5.1 System2-Screenreader_Input	CAPM	MAY	Table 5.58
1.5.2 System2-Screenreader_Output	CAPM	MAY	Table 5.59
1.5.3 System2-Screenreader_Capability	CAPC	MAY	Table 5.60
1.5.4 System2-Screenreader_Processing	CAPP	MAY	Table 5.61
1.6 System2-Magnification	CAPPF	MAY	Like Table 5.57
1.6.1 System2-Magnification_Modality	CAPM	MAY	Like Table 5.57
1.6.2 System2-Magnification_Capability	CAPC	MAY	Like Table 5.60
1.6.3 System2-Magnification_Processing	CAPP	MAY	Table 5.62

5.2.4 System with Additional Software and Hardware AT Support

5.2.4.1 System with Software and Hardware AT Support Description

A desktop computer, like the "system with some additional software AT support" described above, has additional hardware AT (i.e., a HeadMouse Extreme) produced by third party vendors. This system also supports an onscreen keyboard. This system may not be "super accessible" to anyone, but does have the capability of meeting several very specialized needs.

The system has the following:

Input hardware:

- Standard QWERTY keyboard and software that can read this input.
- Microphone and software that can record or transmit via internet this input.
- HeadMouse® Extreme and supporting software.

Output hardware:

- 17inch monitor capable of supporting all standard resolutions and software that supports this display.
- Stereo speakers and software that outputs various types of audio.

Thus, this system's CAP must at least document the following capabilities:

Auditory Input: The presence of a microphone suggests that this system supports auditory input capabilities. Its software limits the range of these capabilities to general sounds to be recorded and/or transmitted (e.g., over the internet). This means that the system has sound and music media auditory output capabilities, but cannot use this information as input data (e.g., speech recognition).

Auditory Output: The presence of stereo speakers suggests this system supports stereo auditory output capabilities. Its software supports audio playback meaning that the system is capable of playback of any previously recorded audio input. Its software also supports system generated audio output, such as error alarms.

Visual Input: The presence of an infra-red camera (for the use of the HeadMouse) suggests this system supports some visual input capabilities. However, these capabilities are limited specifically to the function of the HeadMouse.

Visual Output: The presence of a monitor supporting all standard resolutions suggests this system supports visual output capabilities. Its software fully supports this display. This means that this system is capable of all forms of visual output via the monitor.

Tactile Input: The presence of a keyboard and the support for mouse emulation hardware (i.e., a HeadMouse) suggests this system supports some tactile input capabilities. Its software fully supports these input devices. This means that the system is capable of using text input via keyboard and generic pointing device input for navigation and manipulation of user interfaces.

Tactile Output: The absence of an explicit tactile output device suggests this system does not support tactile output capabilities.

5.2.4.2 System with Software and Hardware AT Support CAP

This system's CAP_{CF}s document the capability of using specific input and output modalities. We start with the CAP_{IC} for this system, shown in Table 5.64.

Table 5.64: A CAP for a "System with both Additional Software and Hardware AT Support"

CAP$_{IC}$ Identification	
Name	SystemWithAT
Type	CAP$_{SYS}$
Qualifier	MAY
Description	This is the starting point of the system CAP for the System with both additional software and hardware AT support example.
CAP$_{IC}$ Linkages	
Higher-CAP$_{O}$	{<Root, >}
Peer-CAP$_{IC}$	
Lower-CAP$_{IR}$	{<System3-Tactile_Input, >, <System3-Auditory_Input, OR>}
Lower-CAP$_{PF}$	{<System3-OnscreenKeyboard, OR>}
Lower-CAP$_{OT}$	{<System3-Visual_Output, >, <System3-Auditory_Output, OR>}

Now we document all of the input capabilities of this system. Since documentation of the keyboard and microphone are treated in the same manner as the Standard System example above, we will not show it here. Thus, for this example we focus on documenting the HeadMouse.

People who cannot use their hands may use a HeadMouse® Extreme (or similar product) to replace a standard USB computer mouse. The HeadMouse® Extreme is essentially a small camera that measures a user's head movements using an infrared light to track a small disposable target that is placed on their forehead. It operates from the top of a computer monitor, laptop computer, or augmentative communication device (Kotzé, Eloff, Adesina-Ojo, & Eloff, 2004).

When used with mouse button software, positioning the pointer and dwelling for a selectable period of time will perform mouse clicks. Alternately, selections can be performed using an adaptive switch, such as a sip/puff switch, or speech recognition software (Origin Instruments Corporation, 2007). Software to mimic a scroll-wheel does not appear to be available at this time.

The HeadMouse could be documented in terms of its actual components (i.e., the infrared camera) plus any adaptive switches. However, because the HeadMouse is based on body movements, especially the head (i.e., it uses proprioception), it still uses the tactile modality, even though it is dependent on the visual modality (i.e., the infra-red camera) to function. This is similar to an optical mouse, which the CAP considers a tactile modality device, even though it uses a laser (or light-emitting diode) and photodiodes to detect movement relative to the underlying surface (Crane & Hastings, 2001). Both are tactile devices, because the user interacts with it tactilely.

Since the CAP is focused on accessibility, it only documents whether the system supports mouse clicking not how this is fully accomplished. Whether the user uses a wired adaptive switch (direct connection) or a wireless adaptive switch (with infrared transmitter), is beyond the scope of the CAP. Thus, although consumers can use wireless connections for their adaptive switches, the CAP may only document the presence of the switches.

Thus, since it is essentially a USB mouse, the HeadMouse is documented within a CAP as a standard mouse rather than as an AT. Treating the HeadMouse like any other mouse means its documentation is largely treated in the same manner as the Standard System example above, as shown in Table 5.65.

Table 5.65: A HeadMouse is Just a Mouse

Identification	
Type	CAP$_{IR}$
Name	System3-Tactile_Input
Qualifier	SHALL
Description	A two-button mouse.
Linkages	
Higher-CAP$_{IC}$	{<SystemWithAT, >}
Peer-CAP$_{IR}$	{<System3-Auditory_Input, OR>}
Peer-CAP$_{PF}$	{<System3-OnscreenKeyboard, OR>}
Peer-CAP$_{OT}$	{<System3-Visual_Output, >, <System3-Auditory_Output, OR>}
Lower-CAP$_{M}$	{<System3-LeftButton, OR>, <System3-RightButton, OR>}
Lower-CAP$_{C}$	{<System3-HeadMouse_Capability, >}
Lower-CAP$_{P}$	
Communication capabilities	
Capacity	1
Parallel	1
Serial	1
Degree	
Intermittent	
Devoted	

This “mouse” has two “buttons”. Table 5.66 shows the CAP for the left mouse button.

Table 5.66: A HeadMouse can Simulate the Left Mouse Button

Identification	
Type	CAP$_{M}$
Name	System3-LeftButton
Qualifier	MAY
Description	The modality of the left-side mouse button.
Linkages	
Higher-CAP$_{IR}$	{<System3-Tactile_Input, >}
Higher-CAP$_{OT}$	
Higher-CAP$_{PF}$	
Peer-CAP$_{M}$	{<System3-RightButton, OR>}
Peer-CAP$_{C}$	{<System3-HeadMouse_Capability, >}
Peer-CAP$_{P}$	
Modality-Specific Information	
ModalityType	TACTILE
MediaTypes	{`UNKNOWN`}
Language	{<`NONE, NIL`>}

The CAP$_C$ for this mouse might look like Table 5.67.

Table 5.67: HeadMouse Tactile Modality Capabilities

Identification	
Type	CAP$_C$
Name	System3-HeadMouse_Capability
Qualifier	MAY
Description	Additional properties and capabilities of the HeadMouse.
Linkages	
Higher-CAP$_{IR}$	{<System3-Tactile_Input, >}
Higher-CAP$_{OT}$	
Higher-CAP$_{PF}$	
Peer-CAP$_M$	{<System3-LeftButton, >, <System3-RightButton, >}
Peer-CAP$_C$	
Peer-CAP$_P$	
Capability-Specific Information	
Capability name	
Capability instance	
Capability values	

Various capabilities related to the HeadMouse, such as how long a pointer must rest (i.e., dwell) before it is interpreted as a click or double-click action, can also be added to the record in Table 5.67. This ends the description of this system's input capabilities.

Now we document all of the output capabilities of this system. Since documentation of the monitor and speakers are treated in the same manner as the Standard System example above, it will not be shown here. This ends the description of this system's output capabilities.

Since the system also comes with an onscreen keyboard built-in, we need to document the CAP$_{PF}$ of this software feature, as shown in Table 5.68.

Table 5.68: An Onscreen Keyboard CAP

Identification	
Type	CAP$_{PF}$
Name	System3-OnscreenKeyboard
Qualifier	MAY
Description	A standard 102-key PC US English QWERTY on-screen keyboard.
Linkages	
Higher-CAP$_{IC}$	{<SystemWithAT, >}
Peer-CAP$_{IR}$	{<System3-Tactile_Input, OR>, <System3-Auditory_Input, OR>}
Peer-CAP$_{PF}$	
Peer-CAP$_{OT}$	{<System3-Visual_Output, OR>, <System3-Auditory_Output, OR>}
Lower-CAP$_{M}$	{<System3-OnscreenKeyboard_Input, >, <System3-OnscreenKeyboard_Output, >}
Lower-CAP$_{C}$	{<System3-OnscreenKeyboard_Capability, >}
Lower-CAP$_{P}$	{<System3-OnscreenKeyboard_Processing, >}
Communication capabilities	
Capacity	1
Parallel	1
Serial	1
Degree	
Intermittent	
Devoted	

When used in select mode, this software works primarily with pointing device input. Thus, the CAP$_{M}$ in Table 5.69 describes its supported visual modality input.

Table 5.69: Onscreen Keyboard Input Modalities

Identification	
Type	CAP$_{M}$
Name	System3-OnscreenKeyboard_Input
Qualifier	MAY
Description	The modality of the onscreen keyboard inputs.
Linkages	
Higher-CAP$_{IR}$	
Higher-CAP$_{OT}$	
Higher-CAP$_{PF}$	{<System3-OnscreenKeyboard, >}
Peer-CAP$_{M}$	{<System3-OnscreenKeyboard_Output, >}
Peer-CAP$_{C}$	{<System3-OnscreenKeyboard_Capability, >}
Peer-CAP$_{P}$	{<System3-OnscreenKeyboard_Processing, >}
Modality-Specific Information	
ModalityType	VISUAL
MediaTypes	{`TextWritten`, `Picture`}
Language	{<`eng, Latn`>}

Table 5.69 documents the onscreen keyboard's visual modality input capabilities:

- *MediaTypes* indicates use of both written text and graphical icons thus supporting use of a pointer cursor on a graphical user interface.

In addition, since this onscreen keyboard provides text output to other software, the CAP_M in Table 5.70 describes its supported outputs.

Table 5.70: Onscreen Keyboard Output Modalities

Identification	
Type	CAP_M
Name	System3-OnscreenKeyboard_Output
Qualifier	MAY
Description	The modality of the onscreen keyboard outputs.
Linkages	
Higher-CAP_{IR}	
Higher-CAP_{OT}	
Higher-CAP_{PF}	{<System3-OnscreenKeyboard, >}
Peer-CAP_M	{<System3-OnscreenKeyboard_Input, >}
Peer-CAP_C	{<System3-OnscreenKeyboard_Capability, >}
Peer-CAP_P	{<System3-OnscreenKeyboard_Processing, >}
Modality-Specific Information	
ModalityType	VISUAL
MediaTypes	{`TextWritten`}
Language	{<`eng, Latn`>}

If desired, a CAP_C can be added to state specific properties, such as size, and settings, such as whether to use scan mode, for the onscreen keyboard using relevant ISO/IEC 24751 tags and other capabilities. Table 5.71 shows one possible CAP_C for the onscreen keyboard.

Table 5.71: Onscreen Keyboard Capabilities

Identification	
Type	CAP$_{C}$
Name	System3-OnscreenKeyboard_Capability
Qualifier	MAY
Description	Additional properties and capabilities of the onscreen keyboard. Here the length, width, layout, and specific other settings are described.
Linkages	
Higher-CAP$_{IR}$	
Higher-CAP$_{OT}$	{<System3-OnscreenKeyboard, >}
Higher-CAP$_{PF}$	
Peer-CAP$_{M}$	{<System3-OnscreenKeyboard_Input, >, <System3-OnscreenKeyboard_Output, >}
Peer-CAP$_{C}$	
Peer-CAP$_{P}$	{<System3-OnscreenKeyboard_Processing, >}
Capability-Specific Information	
Capability name	Length
Capability instance	1
Capability values	{<610, 610, pixels>}
Capability name	Width
Capability instance	1
Capability values	{<190, 190, pixels>}
Capability name	other-ISO/IEC24751-onscreenKeyboard
Capability instance	0
Capability values	... <onscreenKeyboard> <onscreenKeyboardGeneric> <pointAndClick usage = ‘‘required’’> <switchDelay value = ‘‘0.1’’> </pointAndClick> <keyHeight usage = ‘‘required’’ value = ‘‘0.12’’> <keyWidth usage = ‘‘required’’ value = ‘‘0.12’’> <keySpacing usage = ‘‘required’’ value = ‘‘0.12’’> <sound usage = ‘‘required’’ value = ‘‘true’’> </onscreenKeyboardGeneric> </onscreenKeyboard> ...
Capability name	other-ISO/IEC9995-3-keyboardLayout
Capability instance	1
Capability values	en_US

Table 5.71 documents how this keyboard might be used and the amount of screen space it uses:

- The Length and Width capabilities specify the amount of screen real estate taken up by the keyboard.
- The ISO/IEC 24751 tags relating to the onscreen keyboard state whether the keyboard is in "scan mode" (<`pointAndClick`>), the amount of screen real estate taken up by each key (<`keyWidth`> and <`keySpacing`>) and whether auditory output should accompany any keypress (<`soundusage`>).
- *Capability instance* is set to '1' for those Capabilities already used in this CAP (i.e., to describe the physical keyboard).

Finally, a CAP_P is needed to state that all content is transformed into text (i.e., media transformation). This is shown in Table 5.72.

Table 5.72: Onscreen Keyboards Transform their Input

Identification	
Type	CAP_P
Name	System3-OnscreenKeyboard_Processing
Qualifier	MAY
Description	Additional properties and capabilities of the onscreen keyboard. Here the processing function is described.
Linkages	
Higher-CAP_{IR}	
Higher-CAP_{OT}	
Higher-CAP_{PF}	{<System3-OnscreenKeyboard, >}
Peer-CAP_M	{<System3-OnscreenKeyboard_Input, >, <System3-OnscreenKeyboard_Output, >}
Peer-CAP_C	{<System3-OnscreenKeyboard_Capability, >}
Peer-CAP_P	
Processing-Specific Information	
Capabilities	
Capability name	
Capability instance	
Capability values	
Connectivity	
PassThrough	NO
Transformed	YES
Transformations	
MediaTransformation	YES
LangTransformation	NO

This ends the description of this system's processing capabilities.

At a high level, Table 5.73 describes the structure of this system's CAP.

Table 5.73: Structure of CAP_{SYS} for "System with Software and Hardware AT Support"

Name	Type	Unary Operator	Tables
1.0 AnotherStandardSystem	CAPSYS	MAY	Table 5.64
1.1 System3-Tactile_Input	CAPIR	SHALL	Table 5.65
1.1.1 System3-Keyboard_Modality	CAPM	SHALL	Like Table 5.35
1.1.2 System3-Keyboard_Capability	CAPC	SHALL	Like Table 5.36
1.1.3 System3-LeftMouseButton	CAPM	MAY	Table 5.66
1.1.4 System3-RightMouseButton	CAPM	MAY	Like Table 5.66
1.1.5 System3-HeadMouse_Capability	CAPC	MAY	Table 5.67
1.2 System3-Auditory_Input	CAPIR	MAY	Like Table 5.39
1.2.1 System3-Microphone_Modality	CAPM	MAY	Like Table 5.40
1.2.2 System3-Microphone_Capability	CAPC	MAY	Like Table 5.41
1.3 System3-Visual_Output	CAPOT	SHALL	Like Table 5.42
1.3.1 System3-Monitor_Modality	CAPM	SHALL	Like Table 5.43
1.3.2 System3-Screen_Capability	CAPC	SHALL	Like Table 5.44
1.4 System3-Auditory_Output	CAPOT	MAY	Like Table 5.45
1.4.1 System3-LeftSpeaker_Modality	CAPM	MAY	Like Table 5.46
1.4.2 System3-RightSpeaker_Modality	CAPM	MAY	Like Table 5.46
1.4.3 System3-LeftSpeaker_Capability	CAPC	MAY	Like Table 5.47
1.4.4 System3-RightSpeaker_Capability	CAPC	MAY	Like Table 5.47
1.5 System3-OnscreenKeyboard	CAPPF	MAY	Table 5.68
1.5.1 System3-OnscreenKeyboardInput_Modality	CAPM	MAY	Table 5.69
1.5.2 System3-OnscreenKeyboardOutput_Modality	CAPM	MAY	Table 5.70
1.5.3 System3-OnscreenKeyboard_Capability	CAPC	MAY	Table 5.71
1.5.4 System3-OnscreenKeyboard_Processing	CAPP	MAY	Table 5.72

5.2.5 Summary

These example system CAPs were, by necessity, brief. A system's CAP can contain records to document several other capabilities (e.g., the capabilities of each software application, the capabilities of any printers, and so on).

This ends the discussion on example CAPs for systems. Now we will turn to developing example CAPs for environments.

5.3 Environments

The following discussion lists, in a "narrative" form, various environments that CAP_{ENV}s could describe. These narratives are not always complete and detailed. CAPs, when used to describe systems, users, and environments can be far more detailed.

The following Production Rules are to be applied when creating CAP_{ENV}s:

1. Always assume full support for all modalities and media types unless otherwise stated. Be prepared to document what does work.

2. CAP$_{ENV}$s must use the data values described in Section 3.2.

3. If the ALL keyword is not appropriate (i.e., for ModalityType), consider all of the legal data values. It is possible that one or more of the data values cannot be used. While it may be that some data value will eventually not be included in a specific CAP$_{ENV}$, it must still be considered. Specify each data value that can be used.

4. CAP$_{ENV}$ names need to focus on modalities (auditory, visual, tactile, olfactory).

Where ALL modalities are not supported, the best way to construct an environment's CAP is to separately document every one of the environment's capabilities to support interactions in specific modalities. This approach is shown in the examples discussed below.

5.3.1 The CAP Approach to Environments

The definitions of usability and accessibility discussed in Chapter 1 place special emphasis on the role of the environment. These definitions recognize that an environment's usability can either support or add noise into the interactions between users and systems. Similarly, the UARM model presented in Chapter 2 presents an environment as a potential source of "shortcomings" which impact the interactions between users and systems. In terms of the CAP, an environment directly impacts the capability of a system or a user to receive or transmit information in a specific modality.

For example, a microphone connected to a system is most useful if the environment supports its use. If the microphone is intended only to capture all sound in the environment, then the noisiness of the environment will support this use (assuming the environment is not a vacuum, in which case the lack of air will not support this use). On the other hand, if the intent of using the microphone is to capture sound from a specific source within the environment, then the noisiness of the environment may completely block this use.

This section presents two environment examples expressing CAPs in terms of dual modality capabilities:

- The first example presents a CAP which describes a conference exhibition hall. It highlights the issues raised in an environment with too much auditory noise.

- The second example presents a CAP which describes a darkened kitchen. It highlights the issues raised in an environment with too little lighting and the impact of this on users working in the visual modality.

As noted at the beginning of the chapter, linkages are described in terms of <*cap-name, linkage-type*> pairs and, where the *linkage-type* is not specified, the default value for the record type is to be assumed. In the case of environments, AND is the default *linkage-type*. The *linkage-type*s OR and XOR must be explicitly stated. Recall that AND implies that both operands of the conjunction "are included together" (not necessarily simultaneously) for access.

5.3.2 Conference Exhibition Hall

5.3.2.1 Conference Exhibition Hall Description

This environment is a large ballroom filled with various exhibitors. Each exhibitor is speaking to various individual conference attendees about their products / services. The room is well lit from the ceiling. However, in addition to the loud conversational white noise, a malfunctioning fire alarm intermittently sounds. In addition, users who are also exhibitors may be interrupted at any time by a passing conference attendee asking for further information.

Given the above description of this environment, its CAP_{CF}s could document the following user/system interaction needs:

System Auditory Input: The presence of ambient conversational noise may environmentally impact the system's ability to detect the user's voice depending on the type of microphone used. Thus, there is some impact on the use of speech-recognition input options which should be described in this environment's CAP.

User Auditory Input: The presence of loud conversational white noise and the added distraction of the intermittent fire alarm will impact the user's ability to hear. If the user is an exhibitor, the intermittent presence of a potential customer will also distract the user's attention to the system. Thus, there is impact on a system's auditory output options which should be described in this environment's CAP.

System Visual Input: Since the environment is well lit, system input via visible or infra-red spectra is not environmentally impacted. Although, the ability to establish a line of sight is beyond the scope of the CAP, line of sight is likely impacted by the presence of exhibitor booths requiring systems and users to be oriented closely in space. Thus, there is likely no impact on the use of system cameras (i.e., system's visual input).

User Visual Input: Since the environment is well lit with artificial lighting, the user's sight is not environmentally impacted. Although, the ability to establish a line of sight is beyond the scope of the CAP, line of sight is likely impacted by the presence of exhibitor booths requiring systems and users to be oriented closely in space. Thus, there is little impact on the use of a system's visual output options, if the user and visual output device (e.g., monitor) are close in space. There is equally little environmental impact in the user's ability to see the system and its components.

System Visual Output: If the user is an exhibitor, the intermittent presence of a potential customer will visually distract the user's attention to the system. This may require the system, if it is using visual cues to communicate with the user, to use larger persistent visual cues. There is little environmental impact on the system's visual display.

Tactile Input/Output: Since it is not mentioned, the ability to use tactile/haptic input and output devices is likely not environmentally impacted. Thus, there is no impact on the ability to use tactile/haptic input/output options. Although the ability to orient and manoeuvre within space is generally beyond the scope of the CAP, the presence of the exhibitor booths likely impacts full use of the entire space.

5.3.2.2 Conference Exhibition Hall CAP

Given the above description, we can create an environment CAP specifically for this environment. We start by documenting its CAP_{IC}, shown in Table 5.74

Table 5.74: A CAP for a "Conference Hall"

CAP_{IC} Identification	
Name	ConferenceHall
Type	CAP_{ENV}
Qualifier	MAY
Description	This is the starting point of the environment CAP for the Conference Hall example.
CAP_{IC} Linkages	
Higher-CAP_O	{<Root, >}
Peer-CAP_{IC}	{<ConferenceHall-II, AND>}
Lower-CAP_{IR}	{<ConferenceHall_Auditory_IR, >, <ConferenceHall_Visual_IR, >, <ConferenceHall_Tactile_IR, >}
Lower-CAP_{PF}	
Lower-CAP_{OT}	{<ConferenceHall_Auditory_OT, >, <ConferenceHall_Visual_OT, >, <ConferenceHall_Tactile_OT, >}

Table 5.74 documents the start of this environment's CAP:

- Type is filled with the default entry for an environment CAP,
- Qualifier is "MAY" to indicate that use of this CAP is optional to other CAPs,
- Higher-CAP_O links to the Root CAP_O specified at the beginning of the chapter, and
- The Peer-CAP_{IC} will be discussed later.

Now we need to document the various CFs that describe this environment. To achieve this, we need CAP_{CF}s that describe all of the modality, capability, and processing information that is relevant for each of input (CAP_{IR}), output (CAP_{OT}), and processing (CAP_{PF}).

This environment does seem to handicap interactions that use the auditory modality. Where the system depends on auditory output via speakers, it should limit any attempt to use the auditory modality because the user may not be able to hear or attend to it. Where the system depends on auditory output via headset, it could limit any attempt to use the auditory modality because the user may not be able to attend to it, but it does not have to specifically avoid the auditory modality because the headset ensures that the user can hear it. As will be discussed later, this impact on the auditory modality can also be documented in a CAP.

This environment's CAP documents its full support of most modalities and the limitations it places on the auditory modality. As with several of the user examples above, this is accomplished by documenting each modality separately. Since the auditory modality is our primary concern and we have already shown what the records of other modalities might look like in other contexts, this discussion will focus on the auditory modality. Table 5.75 shows the auditory modality CAP_{IR} for this environment.

Table 5.75: Auditory Input CAP

Identification	
Type	CAP_{IR}
Name	ConferenceHall_Auditory_IR
Qualifier	MAY
Description	The example environment's supported auditory input modalities.
Linkages	
Higher-CAP_{IC}	{<ConferenceHall, >}
Peer-CAP_{IR}	{<ConferenceHall_Visual_IR, >, <ConferenceHall_Tactile_IR, >}
Peer-CAP_{PF}	
Peer-CAP_{OT}	{<ConferenceHall_Auditory_OT, >, <ConferenceHall_Visual_OT, >, <ConferenceHall_Tactile_OT, >}
Lower-CAP_{M}	{<ConferenceHall-AuditoryModality, >}
Lower-CAP_{C}	{<ConferenceHall-AuditoryCapabilty, >}
Lower-CAP_{P}	
Communication capabilities	
Capacity	N
Parallel	N
Serial	N
Degree	
Intermittent	
Devoted	

Recall from Chapter 2 that communications can originate from the environment. In addition, environments support communications that are simultaneously inputs and outputs (i.e., "dual"). For environments, such supported communication modalities are recorded as both an IR and an OT, since whether the direction is dual can be deduced from the CAP_{IC} type (i.e., CAP_{ENV}). For this environment, the dual communication directions of its supported modalities are documented by the presence of both the CAP_{IR} shown above and a similar CAP_{OT}.

According to Table 5.75, the example environment is noisy enough to partially handicap user/system interactions that use the auditory modality, but it does not exclude the auditory modality (which would imply either a CAP_M with a NOT qualifier or a CAP_{ENV} listing all of the modalities except the auditory modality).

Note that, according to Table 5.75, the example environment can have multiple channels in the same modality. In this case, the combination of the intermittent fire alarm as well as the conversational noise of several dozen people, provides this environment with innumerable auditory channels.

To document the noise level in the CAP for this environment we need to add a CAP_C that lists the capabilities of the auditory modality. This is shown in Table 5.76.

Table 5.76: Auditory Modality Capabilities

Identification	
Type	CAP_C
Name	ConferenceHall-AuditoryCapabilty
Qualifier	MAY
Description	Additional properties and capabilities of the environment. Here the current level of noise is described.
Linkages	
Higher-CAP_{IR}	{<Auditory_IR, >}
Higher-CAP_{OT}	{<Auditory_OT, >}
Higher-CAP_{PF}	
Peer-CAP_M	
Peer-CAP_C	
Peer-CAP_P	
Capability-Specific Information	
Capability name	Noise
Capability instance	0
Capability values	{<55, 75, dB>, <70, 90, dB>}

Table 5.76 documents the various sources of noise in the environment:

- *Capability values* is a list of the various noises in the environment. Normal conversational speech about 91 cm (about three feet) from a speaker typically averages 65 dB and ranges from 55 to 75 dB (Manlove, Frank, & Vernon-Feagans, 2001). Fire alarms vary in loudness but tend to be rated to exceed the potential ambient noise levels. The level of volume experienced in the environment depends on proximity to the alarm station.

- The linkages recorded in Higher-CAP_{IR} and Higher-CAP_{OT} denote the dual direction of this environment's modalities.

In general, physical environments are by their nature considered language neutral. This means that a physical environment can support all languages in all modalities, if it supports all of the modalities. Environments that do not in fact support specific language modalities may in fact exclude their use. For example, extremely noisy environments may fully handicap, to the point of exclusion, the use of oral languages such as English, but have no impact on written or signed languages.

Social environments on the other hand are, by definition, constrained by the culture of the environment. Although periods may exist where language may not be perceived/present, social environments are not language neutral. Cultural constraints may also play a role in the use of specific languages. For example, legal requirements to display public signage in specific languages only could impact this public environment.

This might end our discussion of this environment, and would if the user only had access to external speakers. As mentioned above, in this environment, if a user were to use a headset and boom microphone, the auditory modality can in fact be fully used in the interaction between the user and the system. This suggests that there exist more than one environment: that in which external speakers are used, and that in which a headset is used. These kinds of micro-environments are documented using additional environment CAPs. The CAP_{ENV} in Table 5.77 describes improved support to the auditory modality.

Table 5.77: A CAP for Using a Headset in a "Conference Hall"

CAP_{IC} Identification	
Name	ConferenceHall-II
Type	CAP_{ENV}
Qualifier	MAY
Description	A CAP describing the impact of using a headset and boom microphone.
CAP_{IC} Linkages	
Higher-CAP_O	{<Root, >}
Peer-CAP_{IC}	
Lower-CAP_{IR}	{<CH2_Auditory_IR, >}
Lower-CAP_{PF}	
Lower-CAP_{OT}	{<CH2_Auditory_OT, >}

Table 5.77 documents the start of this micro-environment's CAP:

- Type is filled with the default entry for an environment CAP.

In this case, the CAP_{IR} listed in Table 5.77 and its equivalent CAP_{OT} are used instead of the CAP_{IR} and CAP_{OT} listed in Table 5.75. The new CAP_{IR} is shown in Table 5.78.

Table 5.78: Micro-Environment's Auditory Input

Identification	
Type	CAP$_{IR}$
Name	CH2_Auditory_IR
Qualifier	MAY
Description	The supported auditory input modalities if a headset is used.
Linkages	
Higher-CAP$_{IC}$	{<ConferenceHall-II, >}
Peer-CAP$_{IR}$	
Peer-CAP$_{OT}$	{<ConferenceHall_Auditory_OT, >}
Lower-CAP$_{M}$	{<CH2-Auditory_Modality, >}
Lower-CAP$_{C}$	{<CH2-Auditory_Capabilty, >}
Lower-CAP$_{P}$	
Communication capabilities	
Capacity	1
Parallel	1
Serial	1
Degree	
Intermittent	
Devoted	

Note that, for the purposes of this Thesis, in this micro-environment there is only one primary auditory channel (i.e., the headset). Depending on its quality, it is possible that, while wearing the headset, the user can still hear other ambient noise from the outer environment.

The equivalent CAP$_{OT}$ to the record in Table 5.78 is similar and thus not shown here. Table 5.79 shows the CAP$_{M}$.

Table 5.79: Micro-Environment's Auditory Input Modality

Identification	
Type	CAP$_{M}$
Name	CH2-Auditory_Modality
Qualifier	MAY
Description	The modality of the micro-environment.
Linkages	
Higher-CAP$_{IR}$	{<CH2_Auditory_IR, >}
Higher-CAP$_{OT}$	{<CH2_Auditory_OT, >}
Peer-CAP$_{M}$	
Peer-CAP$_{C}$	{<CH2-Auditory_Capabilty, >}
Peer-CAP$_{P}$	
Modality-Specific Information	
ModalityType	AUDITORY
MediaTypes	{`TextSpoken`, `Sound`, `Music`}
Language	{<`eng, NIL`>}

The micro-environment is quieter enough to support user/system interactions that use the auditory modality. To document the noise level in the CAP for this micro-environment we need to add a CAP_C that lists the new capabilities of the auditory modality. This is shown in Table 5.80.

Table 5.80: Micro-Environment's Auditory Input Capabilities

Identification	
Type	CAP_C
Name	CH2-Auditory_Capabilty
Qualifier	MAY
Description	Additional properties and capabilities of the micro-environment. Here the new level of noise is described.
Linkages	
Higher-CAP_{IR}	{<CH2_Auditory_IR, >}
Higher-CAP_{OT}	{<CH2_Auditory_OT, >}
Peer-CAP_M	{<CH2-Auditory_Modality, >}
Peer-CAP_C	
Peer-CAP_P	
Capability-Specific Information	
Capability name	Noise
Capability instance	0
Capability values	{<0, 55, dB>}

Table 5.80 documents the noise in this micro-environment. Note that the headset being used is very good and blocks most of the room noise.

Table 5.81: Structure of "Conference Hall" Environment CAP

Name	**Type**	**Unary Operator**	**Tables**
1.0 ConferenceHall	CAPENV	MAY	Table 5.74
1.1 ConferenceHall_Auditory_IR	CAPIR	MAY	Table 5.75
1.1.1 ConferenceHall-AuditoryModality	CAPM	MAY	
1.1.2 ConferenceHall-AuditoryCapabilty	CAPC	MAY	Table 5.76
1.2 ConferenceHall_Visual_IR	CAPIR	MAY	
1.3 ConferenceHall_Tactile_IR	CAPIR	MAY	
1.4 ConferenceHall_Auditory_OT	CAPOT	MAY	
1.1.1 ConferenceHall-AuditoryModality	CAPM	MAY	
1.1.2 ConferenceHall-AuditoryCapabilty	CAPC	MAY	
1.5 ConferenceHall_Visual_OT	CAPOT	MAY	
1.6 ConferenceHall_Tactile_OT	CAPOT	MAY	
1.7 ConferenceHall-II	CAPENV	MAY	Table 5.77
1.7.1 CH2_Auditory_IR	CAPIR	MAY	Table 5.78
1.7.1.1 CH2-Auditory_Modality	CAPM	MAY	Table 5.79
1.7.1.2 CH2-Auditory_Capabilty	CAPC	MAY	Table 5.80
1.7.2 CH2_Auditory_OT	CAPOT	MAY	Like Table 5.78
1.7.1.1 CH2-Auditory_Modality	CAPM	MAY	Like Table 5.79
1.7.1.2 CH2-Auditory_Capabilty	CAPC	MAY	Like Table 5.80

This ends the description of this environment's capabilities. At a high level, Table 5.81 describes the structure of this environment's CAP.

Note: The missing tables in Table 5.81 are most analogous to the user examples presented in Section 5.1.

5.3.3 A Darkened Kitchen

5.3.3.1 Darkened Kitchen Description

A medium-sized kitchen with no lights turned on. Most nearby rooms also have no lights on to ensure that the bedrooms have as little light as possible (so that the children in them can sleep). Ambient lighting is available through a nearby window (however, it is dark outside) and the light from the door of a nearby room. Further ambient lighting is provided by the system's screen and backlit keyboard. The house is relatively quiet.

Given the above description of this environment, its CAP_{CF}s could document the following user/system interaction needs:

System Auditory Input: Since the space is quiet, there is little chance ambient noise will impact the system's ability to detect the user's voice and other environmental sounds. Thus, there is no impact on speech-recognition system input options.

User Auditory Input: The quiet of the room will not environmentally impact the user's ability to hear. Thus, there is no impact on a system's auditory output options such as text-to-speech output and system alarms.

System Auditory Output: Because quiet is desired, the system may need to reduce the volume of its auditory output. There is some impact on system audio output options.

User Auditory Output: Because quiet is desired, the user may choose to reduce the volume of their voice or not speak altogether. Thus, there is some impact on a user's ability to use system input options such as speech-recognition.

User Visual Input: Since the environment is poorly lit, sight is somewhat environmentally impacted. Although beyond the scope of the CAP, the presence of screen lighting and a backlit keyboard reduces some of the chance that the user's ability to see the system is environmentally impacted. However, the ability to see other elements of the task (e.g., nearby papers) may be handicapped by this environment. Thus, while there is no impact on the use of a system's visual output options, there is environmental impact in the user's ability to see the system and its components for which the system needs to accommodate. This needs to be described in this CAP.

System Visual Input: Since the environment is poorly lit, sight via visible spectrum alone is environmentally impacted. System input via the infra-red spectrum is not environmentally impacted. Thus, there is some impact on the use of system cameras which should be described in this CAP.

Tactile Input/Output: The ability to use tactile/haptic input and output devices is likely not environmentally impacted, since it is not mentioned otherwise. Although, the ability to orient and manoeuvre within space is generally beyond the scope of the CAP, full use of this environment is likely possible, since it is not mentioned otherwise.

Olfactory Input/Output: Given this environment is a kitchen with potential for food smells and other olfactory stimuli, olfactory noise could well be present. However, since none are mentioned, there does not appear to be an olfactory irritant or distracter in the environment.

5.3.3.2 Darkened Kitchen CAP

Given the above description, we can create an environment CAP specifically for this environment. We start by documenting its CAP$_{IC}$, as shown in Table 5.82.

Table 5.82: A CAP for a "Darkened Kitchen"

CAP$_{IC}$ Identification	
Name	Kitchen
Type	CAP$_{ENV}$
Qualifier	MAY
Description	This is the starting point of the environment CAP for the Darkened Kitchen example.
CAP$_{IC}$ Linkages	
Higher-CAP$_{O}$	{<Root, >}
Peer-CAP$_{IC}$	{<Kitchen-Channel, AND>}
Lower-CAP$_{IR}$	{<Kitchen_Visual_IR, >, <Kitchen_Auditory_IR, >, <Kitchen_Tactile_IR, >}
Lower-CAP$_{PF}$	
Lower-CAP$_{OT}$	{<Kitchen_Visual_OT, >, <Kitchen_Auditory_OT, >, <Kitchen_Tactile_OT, >}

Now we need to document the various CFs that describe this environment. To achieve this we need CAP$_{CF}$s that describe all of the modality, capability, and processing information that is relevant for each of input (CAP$_{IR}$), output (CAP$_{OT}$), and processing (CAP$_{PF}$).

This environment's CAP documents its full support of all modalities and the temporary lighting-related issues.

For this environment, separately documenting support for all modalities but visual is achieved in the same manner as the Conference Hall environment example described above and will not be repeated here. This discussion will focus on the visual modality.

The environment's lighting is a visual modality that is both an input to and an output of the environment (i.e., dual). Since the environment's lighting is dual, CAP_{IR} and CAP_{OT} need to be provided. This is achieved in the same way as the auditory inputs and outputs were documented in the Conference Hall example above and thus not shown here.

Given an environment's lighting is a visual modality; the CAP_M would document this modality. This is achieved in the same way as above and thus not shown here.

As shown in Table 5.83, the temporary lighting-related issues are documented in terms of a CAP_C on the environment's lighting CAP.

Table 5.83: Illumination of the "Darkened Kitchen"

Identification	
Type	CAP_C
Name	Kitchen-VisualCapabilty
Qualifier	MAY
Description	Additional properties and capabilities of the kitchen. Here the current lighting is described.
Linkages	
Higher-CAP_{IR}	{<Kitchen_Visual_IR, >}
Higher-CAP_{OT}	{<Kitchen_Visual_OT, >}
Higher-CAP_{PF}	
Peer-CAP_M	{<Kitchen-VisualModality, >}
Peer-CAP_C	
Peer-CAP_P	
Capability-Specific Information	
Capability name	Illumination
Capability instance	0
Capability values	{<20, 50, lx>}

The SI unit lux (abbreviated 'lx') is a measure of illumination. A kitchen is normally about 500 lx (Charness & Dijkstra, 1999).

Should the lighting of this environment change due to, for example, the lights being turned on, this would impact this environment's CAP only in that it would remove any need for the CAP_C in Table 5.83, as well as enable describing the environment in terms of a CAP that is allowed to use the keyword ALL (i.e., an "ALL-Modalities" oriented environment CAP).

This ends the description of this environment's capabilities.

At a high level, Table 5.84 describes the structure of this environment's CAP.

Table 5.84: Structure of "Darkened Kitchen" Environment CAP

Name	Type	Unary Operator	Tables
1.0 Kitchen	CAPENV	MAY	Table 5.82
1.1 Kitchen_Visual_IR	CAPIR	MAY	
1.1.1 Kitchen-VisualModality	CAPM	MAY	
1.1.2 Kitchen-VisualCapabilty	CAPC	MAY	Table 5.83
1.2 Kitchen_Auditory_IR	CAPIR	MAY	
1.3 Kitchen_Tactile_IR	CAPIR	MAY	
1.5 Kitchen_Visual_OT	CAPOT	MAY	
1.1.1 Kitchen-VisualModality	CAPM	MAY	
1.1.2 Kitchen-VisualCapabilty	CAPC	MAY	
1.6 Kitchen_Auditory_OT	CAPOT	MAY	
1.7 Kitchen_Tactile_OT	CAPOT	MAY	

Note: The missing tables in Table 5.84 are most analogous to the user examples presented in Section 5.1.

5.3.4 Summary

These example environment CAPs were, by necessity, brief. An environment's CAP can also contain records to document several other capabilities. For example, the Kitchen environment could be described using several CAP_{ENV}s for the visual modality to document various different lighting levels, depending on time of day and sets of artificial lighting used.

This ends the discussion on example CAPs for environments.

5.4 Next Steps

This chapter shows that CAPs can be generated using the tables developed in Chapter 3 and the binary and unary relationships developed in Chapter 4. The next chapter will discuss the validation of these results.

Chapter 6

Validation

This chapter discusses the validation of current CAP research. Section 6.1 discusses the need for both validity and reliability for the CAP. Section 6.2 discusses the ISO approach to reliability and presents further discussion on validating the CAP. Section 6.3 discusses appropriate metrics for the CAP and some considerations of the theoretical and practical issues needed for developing CAP-based metrics.

6.1 Validity and Reliability

Usability, as previously discussed (ISO, 1998a), involves effectiveness, efficiency, and satisfaction. Two important concerns relating to effectiveness are validity and reliability. Validity occurs if the CAP measures what it is supposed to and reliability occurs if it yields consistent results. Although the CAP could be reliable but not valid, it cannot be valid without first being reliable. Validity cannot be assumed no matter how reliable the CAP is.

A profile is a composite structure of meaningful properties or attributes. Metrics based on profiles can take two forms:

- a structure containing several attributes which may or may not interact with each other, but who together describe a specific dimension under evaluation (e.g., the so-called "Profile Metric" uses a profile of attributes to evaluate software complexity, see McQuaid, 1996), (i.e., an instance of the profile itself) or

- comparison of two or more instances of the profile to generate a metric of similarity (or a metric of difference) that is the basis of evaluation (e.g., "Ringo" is an expert system based on a profile of the user's music interests (positive as well as negative), that is compared and weighed against the profiles of other users to recommend (or advise against) items to the user, see Shardanand & Maes, 1995).

The CAP is a metric in both senses. It is a structure of attributes, which separately look very different and may or may not interact with each other, but together describe the same dimension (i.e., the needs of a user, system, AT, or environment). It is also designed for comparison of two or more CAPs to evaluate the accessibility of a specific context or situation. The CAP is not a vector metric — while CAPs are multidimensional, they do not render a single number result. This will be discussed later.

For the purpose of this thesis, **measurement** is a process, based upon the rules of a specific method or "metric" (e.g., survey, test, ruler, interview, etc.), that captures information about an entity. For the purpose of this thesis, a **measure** is the result of measurement, it is the symbol assigned to the entity under evaluation to characterize the attribute (Fenton & Pfleeger, 1997). This process and its results can be qualitative, quantitative, or both[1].

6.1.1 Validity

Preferably, a CAP should be **valid** in that it should represent what it claims to represent. Valid metrics tell us something about the attribute being evaluated. Thus, what needs to be valid is the meaning or interpretation of any evaluation(s) based on the CAP, as well as the implications for action resulting from this meaning (Messick, 1995).

There are many different types of validity, which all have to do with the different threats and biases, which would undermine the meaningfulness of any metric (Garson, n.d.). Many of the various types of validity overlap. The following is a discussion of four major types of validity: face, content, criterion-oriented (including predictive, concurrent, and experimental validity), and construct.

6.1.1.1 Face Validity

Often used as an initial proof of concept, face validity usually implies asking a colleague or expert in the field to vouch for the items measuring what they were intended to measure. Face validity is simply an assertion that the metric reasonably measures what it is intended to measure. It is essentially "take my word for it" validity (O'Connor, 2004).

For example, for the CAP to become an International Standard, it must pass a vote of international experts. At this point, the basis of this research (i.e., Fourney, 2004) has already achieved face validity by receiving the vote of experts from six countries. As will be discussed in Section 6.2, this research has nearly completed the last stage in becoming an International Standard through thorough multiple peer reviews, which consider face and other forms of validity and reliability.

[1]An example of both is human perception. The human eye can measure the quantitative frequency of the light that reflects off of a "blue" wall, but a colour-blind observer (with no damage to their retinal cone cells) might qualitatively interpret this result differently from another observer.

6.1.1.2 Content Validity

The act of judging the adequacy of content for a potential measure is called content validation. Content validity is based on the ideas of conceptualization and operationalization. Conceptualization is taking a construct or concept and refining it by developing a conceptual or theoretical definition. Operationalization is taking a conceptual definition and making it more precise by linking it to one or more specific, concrete indicators or operational definitions (O'Connor, 2004). If a researcher has focused in too closely on only one type or narrow dimension of a construct or concept, then it is conceivable that other indicators were overlooked. As a result, the metric may lack content validity. Thus, content validity is making sure the conceptual space is fully covered (Munson, 2003). Unfortunately, content validity assumes the existence of a good detailed description of the content domain. This is not always the case.

Content validity has been established for the CAP structure. Conceptualization occurred in Chapter 2 with the development of the UARM model into the CAP model. Operationalization began with the initial specification of the CAP developed in Chapter 3 and its operators defined in Chapter 4. It continues as a direction for future research.

6.1.1.3 Criterion-Oriented Validity

Criterion validity is based on correlation rather than explanation — one only really knows good criterion attributes when they are seen to work in real life. There are several types of criterion-oriented validity, three are: predictive validity, concurrent validity, and experimental validity.

Predictive validity has to do with how well a measure of something accurately predicts some event in the future. Typically, this involves speculating that some attribute A of an object, which can be well-measured, is a good predictor of the presence of attribute B for which a metric is being tested. Both the metric under consideration and the known metric are used to evaluate known objects, which exhibit and exemplify both attributes being measured and to evaluate less-known objects for whom one knows they exhibit attribute A, but it is not clear if they exhibit attribute B. Comparison of the results shows whether the new metric might validly measure attribute B (Munson, 2003).

For example, it is intended that consumers will be able to compare CAPs to aid the selection of products to ensure an accessible system. For CAPs to show predictive validity (i.e., predicting "accessibility"), it should be shown that the various attributes recorded in a CAP are, as a whole, good predictors of the presence of accessibility. Theoretically, the CAP shows predictive validity as a result of the discussion in Chapter 3. However, full predictive validity is only possible with the full implementation of the CAP and extensive experimental investigation, which is beyond the scope of this Thesis.

Concurrent validity is concerned with how well something estimates actual day-by-day behaviour. Typically, the metric under consideration is used to evaluate known objects which exhibit and exemplify the attribute being measured, and to evaluate less-known objects for whom one wants to know beforehand if they exhibit the attribute. Comparison of the results shows whether the metric might validly measure the attribute (Munson, 2003).

For example, an assessment of whether the CAP of a system that is known to be accessible should differ significantly and appropriately from the CAP of a system with known accessibility problems. For the difference to be appropriate, it must relate to accessibility.

Experimental validity is concerned with both the extent to which an experiment rules out alternative explanations (i.e., internal validity) and whether it applies to other individuals, settings, and conditions (i.e., external or ecological validity). Typically, the metric under consideration is used in a formal experimental study of the attribute being measured. The results of the study show whether there are other explanations to consider within the model and/or whether the metric can be generalized.

Establishing these types of validity is beyond the scope of this Thesis. Predictive validity takes a long period to establish. Concurrent validity requires access to a range of real users and systems. Experimental validity requires access to a sophisticated lab. Establishing these types of validity should be considered as an area for future research.

6.1.1.4 Construct Validity

Not all metrics represent what their names imply. Each metric seeks to qualify and/or quantify some underlying construct. Construct validity is measured by the correlation between the intended independent variable (construct) and the proxy independent variable (indicator, sign) that is actually used (Munson, 2003). Thus, construct validity seeks agreement between a theoretical concept and a specific metric. It is the most difficult validity to achieve, as it can take years of research to investigate.

Construct validity can be further separated into two sub-categories: convergent validity and discriminate validity. **Convergent** validity describes how well various metrics, which should be theoretically related, show general agreement. **Discriminate** validity is the idea that metrics which theoretically should not be related clearly show a lack of a relationship.

To quickly achieve construct validity, a theoretically-related metric of accessibility such as metrics based on the comparison of CAPs would need to be evaluated against some other accessibility evaluation metric. However, even if a candidate accessibility evaluation metric exists to compare CAP-based accessibility metrics with, the comparison of CAPs is beyond the scope of this Thesis. Thus, the construct validity of the CAP is beyond the scope of this Thesis.

6.1.2 Reliability

CAPs cannot be valid, unless they are in some way considered **reliable**. Reliability is the extent to which a CAP yields the same result on repeated trials. If a metric is not at least somewhat reliable, the information gleaned from measuring the attribute is of no value. Thus, if CAPs were not reliable, there would be no way to use them as a basis for conclusions, to formulate theories, or to claim generalizability.

The question asked in reliability is: Do different judges looking at the same attribute assess it in the same manner (Munson, 2003)? Two general classes of reliability are: Test-retest reliability and Inter-rater reliability. Establishing reliability requires extensive research and time; thus, establishing these types of reliability should be considered as an area for future research.

6.1.2.1 Test-Retest Reliability

Test-retest reliability can be used to determine the consistency of a metric from one time to another. Test-retest reliability is important because, if one can apply the same metric to the same entities at two significantly different points in time with as close to similar results as possible, the stability of the metric can be assessed over time and the expectation that the metric can produce consistent results is supported (Garson, n.d.).

The amount of time between tests is crucial. Since test-retest reliability assumes that there has been no substantial change over time in the construct being measured, the appropriate length of the interval between tests depends on the stability of the attributes being measured (Garson, n.d.). If the attributes being measured are not expected to change over a very long period of time, then a longer interval is needed. If the attribute being measured changes frequently, then a rather short interval is needed.

For example, an evaluation of the accessibility of a given system could be performed with a CAP. This evaluation could then be repeated after an appropriate amount of time (likely several months). Assuming that the user or the system is not changed in any way (e.g., by software updates), comparison of the results would provide a reasonable estimate of the reliability of the CAP.

6.1.2.2 Inter-Rater Reliability

Inter-rater (or Inter-observer) reliability is used to assess the degree to which different raters/observers give consistent estimates of the same entity. This involves administering the same metric to the same entity by two or more observers to establish the extent of consensus on use of the metric. Presumably, observers should not know the expected outcomes of the use of the metric (Garson, n.d.). Inter-rater reliability is important because it supports the expectation that the metric can produce consistent results, while at the same time showing that it is usable by others.

For example, an evaluation of the accessibility of a given system could be performed with a CAP. This evaluation would be repeated by several trained observers. Comparison of the results would provide a reasonable estimate of the reliability of the CAP. This estimate of reliability takes somewhat less time to perform when compared to test-retest reliability.

6.1.2.3 Expert Evaluation of Reliability

Related to Inter-rater reliability is the use of expert opinion as a basis for *estimating*, as opposed to establishing, the reliability of something. This is essentially the approach used in academic peer review.

While reviewers might agree (or disagree) on ratings of "accept", "reject", or "resubmit" (i.e., their inter-rater reliability or even intra-rater reliability), the points of a publication under review on which they focus may differ substantially (Campanario, 1998). That is, their expert opinion may be similar, but for different reasons.

This is discussed in greater detail below.

6.2 Evaluating the CAP via ISO/IEC Expert Voting

The CAP has been subjected to rigourous external expert evaluation, because it is the basis of an ISO/IEC standard.

International standards play an important role in global commerce. One of the major organizations involved in their development is ISO. ISO's documents are developed by working groups — committees of people from around the world with technical expertise relating to the subject matter of its activities / documents. The ISO/IEC development process involves multiple nations participating in both the development and review of a document through a five-stage process (Working Draft, Committee Draft, Final Committee Draft, Final Draft International Standard, and International Standard). At each stage, approval by all interested members of the committee developing it must occur before a document can move to the next stage (International Organization for Standardization [ISO] and International Electrotechnical Commission [IEC], 2004).

"Approval" is achieved through a voting process called "balloting", where each country that is a voting member of the committee has one vote. Member countries can vote "Yes" or "No" to approve or disapprove a document. Each country's vote can also include comments to improve/correct and further develop the document. Votes can take from two to six months depending on the document's development stage.

A new work item can be proposed at any time to a relevant committee. This proposal is circulated for ballot to determine whether sufficient interest exists within the Working Group to work on the new document.

Once the document has reached a sufficient point of maturity, it can be distributed for ballot as a Committee Draft. This is the first opportunity for other member countries to review the work being done by the committee.

Once comments generated during the Committee Draft vote have been resolved and further development has occurred, the document can be disseminated for ballot as a Draft International Standard (also known as a "Final Committee Draft" or FCD).

Once comments generated during the Draft International Standard vote have been resolved and further development has occurred, the document can be sent out for ballot as a Final Draft International Standard. This is the final stage and, once any new comments are resolved, this document is then published by ISO as an International Standard.

The process of moving a document into a standard can take up to five years. In addition, after a standard is produced, it is reviewed every five years.

This process of expert evaluation can involve dozens of experts across a minimum of five countries (and generally considerably more) reviewing the same document at once. This form of expert review is a qualitative method for establishing validity (both content and face validity), as well as predicting reliability. Passing a peer-review process is often considered in the scientific community to be a certification of validity. Because of the number of ballots and the number of experts involved, the ISO review process is a more intensive process than typical academic peer review. While academic peer review may not focus on a detailed assessment of reliability, the ISO process focuses on reliability because the national votes agree (or disagree) not only on whether or not a document should be published, but also that, if it is published, then it should be used. To be able to be used, the standard had better be reliable.

Standards present international agreements on reasonable expectations in a wide range of fields. Because of their widely respected status, international standards may even provide the basis for national and international legislation. The development of international accessibility standards is an important step in achieving universal access. ISO/IEC JTC1 SC35 "User Interfaces" has recognized the potential of the CAP and is developing ISO/IEC 24756 "Information technology – Framework for specifying a common access profile (CAP) of needs and capabilities of users, systems, and their environments" based on it (ISO & IEC, 2007a). For more information see Appendix E.

As noted in Chapter 3, ISO/IEC 24751, "Individualized adaptability and accessibility in e-learning, education and training" (currently an FDIS) is a multi-part standard intended to facilitate the matching of individual user needs and preferences with educational resources that meet those needs and preferences (ISO & IEC, 2006e, 2006c, 2006d). Like the CAP, it recognizes that such mismatches can be caused by any number of circumstances such as requirements related to client devices, environments, language proficiency or abilities. The purpose of ISO/IEC 24751 is not to point out flaws in educational resources with respect to accessibility and adaptability, but rather to facilitate the discovery and use of the most appropriate content components for each user (ISO & IEC, 2006e, 2006c, 2006d). The CAP can be supplemented by ISO/IEC 24751 to provide more detailed accessibility requirements for use in domains, such as education, where developers may be expected to support these more detailed requirements. Further, the CAP can provide a platform upon which the ISO/IEC 24751 specification can be generalized to computing rather than specifically to e-learning and thus support capabilities not already realized within the CAP.

The work of this Thesis was accepted in the CD ballot of ISO/IEC 24756 (eight of ten countries accepted). Furthermore, for each nation involved in this voting process, on average, five national experts contributed to each vote. This suggests that the experts in these countries believe that the CAP is valid and will be reliable, and expect it will be used. The issues raised in that ballot were reflected back into this Thesis as well as the standard. The revised version of ISO/IEC 24756, then passed an FCD ballot. Having been expanded based on further research for this Thesis, ISO/IEC 24756 is currently undergoing a second FCD ballot. Within the ISO five-stage process, ISO/IEC 24756 should be published as an International Standard by the end of 2008. This process provides an additional approach to the validity and peer review of this Thesis.

While this voting process provides a preliminary form of validation, actual implementations of the CAP will need to be tested to confirm it. Furthermore, such testing will need to be on-going to strengthen the validity of the CAP.

6.3 Metrics

According to ISO/IEC 14598-1, a metric is a defined measurement method and scale (International Organization for Standardization [ISO] and International Electrotechnical Commission [IEC], 1998a). Traditionally, metrics measure constructs in terms of single values (that are preferably single numbers), but the CAP is a very different kind of metric because it is a structure with multiple components each of which can contain one or more attributes. While some of the attributes documented by CAPs may be described in terms of vector metrics, CAPs themselves are not vector metrics — there is no one "magic" number.

Various metrics are used in one's everyday life. Consider for example:

- the litre is a metric which might be used to measure the amount of fuel purchased for a vehicle or the amount of liquid in a bottle.
- the colour of a room is an attribute that can be measured, so in a room with a blue wall one can say "blue" is a measure of the colour of the room (Fenton & Pfleeger, 1997).
- human attributes such as intelligence might be measured using an IQ test (Mackintosh, 1998).

- clothing size might be labelled “small” or “large”

Metrics include methods for categorizing both qualitative and quantitative data (ISO & IEC, 1998a). As the above examples show, when measuring an entity, categories like “small” and “large” or numbers like “ten litres” can be used. This means that applying a metric to directly or indirectly measure an entity’s attributes implies first having some intuitive understanding of that entity and its attributes (Fenton & Pfleeger, 1997).

This intuitive understanding is relatively easy, if the entity is both known to exist and concrete. For example, a ruler can be used to directly measure the length of a wall. Similarly, a “blue” wall might be evaluated using some mechanism that can measure the frequency of the light that reflects off the wall (e.g., the human eye). However, when an entity is more “fuzzy” or more difficult to directly measure, the task of both deciding what metric to use and assigning a measure to the attribute becomes more difficult. It may be necessary to have a model of the entity under consideration in mind, before even trying to explore ways to measure it. Such a model might suggest indirect ways to measure the desired attribute such as other closely related attributes that are easier to measure, or the model might correct a perception that a particular attribute is even necessary to measure. For this Thesis the CAP is the model that plays this role.

This section discusses how international standards typically approach metrics and describes how the CAP approaches metrics. For more general information on metrics, see Appendix F.

6.3.1 The Standards Approach to Metrics

According to ISO 14598-1, software product evaluation has two objectives: to identify problems so that they can be rectified, and to compare the quality of a product with alternative products or against requirements (which may include certification) (ISO & IEC, 1998a). The definition of many software quality characteristics does not easily allow direct measure. If software product metrics are to be used easily and economically, there is a need to establish metrics that correlate to the characteristics of the software product (ISO & IEC, 1998a). Examples of these characteristics described by ISO 9126-1 include: functionality, reliability, usability, efficiency, maintainability, and portability (International Organization for Standardization [ISO] and International Electrotechnical Commission [IEC], 2001a). Accessibility should be added to this list.

Metrics can differ depending on the environment and the phases of the development process in which they are used. Further, the type of metric required depends on the purpose of the evaluation. If the primary purpose is to understand and correct deficiencies, several metrics may be used to monitor and control improvements. Useful metrics for these purposes include checklists and expert opinion. The primary requirement is that the metric used correctly identifies the impact that any changes in the software will have on quality (ISO & IEC, 1998a).

Valid comparison requires measurement. More rigourous metrics are required to make reliable comparisons, either between products or with criterion values.

Metrics should also adhere to the goals of validity and reliability. Internal metrics should have predictive validity, that is they should correlate with some desired external criterion. Allowing for possible measurement errors caused by measurement tools or human error, a metric should measure the software quality characteristic (or subcharacteristic) it claims to be measuring with sufficient accuracy to allow criteria to be set and comparisons to be made (ISO & IEC, 1998a).

6.3.1.1 Approaches to Usability Metrics in Standards

As mentioned in Chapter 2, the CIF provides a standard means to report the results of usability tests (ISO & IEC, 2006a). According to the working definition of accessibility for this Thesis, accessibility is the usability of a product, service, environment or facility by people with the widest range of capabilities (ISO, 2003b). This suggests that a standard on reporting usability should have something to say about reporting accessibility.

Surprisingly, the CIF says little about reporting or measuring accessibility. Reports that comply with the CIF are required to provide a "full product description." This information must include a description of, "Any groups with special needs that are supported by the product." (ISO & IEC, 2006a). There is no further discussion as to whether this requirement applies to all products or only those that are specifically designed for users with disabilities. Given the American origins of this standard and *U.S. Public Law 105-220, Section 508* (U.S. Federal Government, 1998), which mandates the accessibility of information technologies, this description should apply to all products; however, the wording of the requirement would suggest that it is only intended for specialized software.

The CIF outlines a way to report metrics of efficiency, effectiveness, and satisfaction (ISO & IEC, 2006a). It does not make any suggestions for reporting accessibility, nor does it point to any standards that might assist the report writer in measuring accessibility. This suggests that the CAP, if it is compatible with the CIF, may have a role to play within the CIF.

The CIF has several reporting requirements for information which is already included within the CAP. For example:

- Within the "full product description" mentioned above, there is also the requirement to provide a, "Brief description of the environment in which it should be used." (ISO & IEC, 2006a). Further, when describing the context of product use in the test, the CIF requires a description of the test participant's computing environment (including display devices, audio devices, and input devices). Through the use of environment ICs, which describes the user's physical environment, and system ICs, which describe the user's overall computing environment, both requirements of the CIF can be met by the CAP (although in much greater detail than is required).

- The CIF requires a description of the participants tested. This information can be provided in a table describing the age, gender, education, occupation, professional experience, computer experience, product experience, and special needs of each participant (ISO & IEC, 2006a). Although most of this information goes beyond that contained in a CAP_{USE}, the data on user capabilities contained within a CAP_{USE} is lower-level information that falls into several of these areas.

Although a CAP provides a greater level of detail than is demanded by the CIF, it does provide information that is reportable within a CIF compliant test report. Thus, the CAP is compatible with the CIF and has a role to play within the CIF, because the CIF is at a higher level of abstraction.

With the exception of a section on usability metrics, the CIF has little to say about metrics themselves. Given the definition of usability in ISO 9241-11, CIF compliant reports are required to provide information regarding effectiveness, efficiency, and satisfaction. Any measures of effectiveness and efficiency must be reported, "even when they are difficult to interpret within the specified context of use" (ISO & IEC, 2006a). If the resulting measure of effectiveness or efficiency is meaningless, the report must specify why. Section 2.1.1.1 notes several commonly used metrics for effectiveness, efficiency, and satisfaction.

The CIF does not specify which specific metrics must be used. A CIF compliant report can use any metric it chooses. However, to support combination and comparison, the CAP requires a common set of metrics.

Finally, the CIF has little to say about using metrics. A CIF compliant test report is required to include a description of the metrics used in a usability experiment. This description includes an operational definition of the metric.

As part of a program of work suggested by ISO/IEC JTC1/SC7, work is currently underway to add other "Common Industry Formats" for usability-related purposes (e.g., usability requirements and other usability-related information) and non-usability information (e.g., software quality attributes and test reports) within a Joint Working Group of ISO/IEC JTC1/SC7 and ISO TC159/SC4 (International Organization for Standardization [ISO] and International Electrotechnical Commission [IEC] SC7 Study Group, 2005; International Organization for Standardization [ISO] and International Electrotechnical Commission [IEC], 2006f, 2006g).

6.3.2 The CAP's Approach to Metrics

To use the CAP for the evaluation of accessibility, there is a need to compare CAPs of different or similar types. Chapter 4 focused on combining CAPs of the same type within a larger CAP_{IC} or

CAP_O.

CAPs may be compared CAP attribute-to-CAP attribute (e.g., *MediaType* to *MediaType*) or as a whole. When comparing CAP attribute-to-CAP attribute, the setting of a CAP attribute in one CAP is compared against the setting of the same CAP attribute in another CAP. Similarly, when comparing whole CAPs, the cumulative result of algorithms that compare each CAP attribute is used. Since a CAP is not a vector metric of accessibility, comparison of CAPs only makes sense in terms of a resulting list of incompatibilities.

This subsection discusses what is needed to enable comparison of CAPs. First, the need for a standard dictionary of capabilities is discussed. Second, issues related to the operators needed to compare CAPs are discussed. Finally, since metrics should display some precision, the precision of CAP-based metrics is discussed.

6.3.2.1 Need for Standardized Capability Names

One major issue affecting any future meaningful comparison of CAPs is a need for a standardized set of *Capability names*. When comparison of capabilities is being performed, information related to capabilities will be compared. Currently, there is no guarantee that the names of the capabilities in one CAP record match those of another. For example, the *Capability name* "Force" might be used in one CAP and "Strength" in another. It is vital that the name of comparable capabilities be equivalent to ensure that matching on each capability can be completed. Because of its importance in the comparison process, development of a standardized *Taxonomy of Capabilities* must occur before defining algorithms to compare CAPs. This would require a review of relevant documents and standards to identify all possible *Capability names*, listing these names in some logical order, and then defining the purpose / meaning of each name. This requirement is necessary to conduct CAP attribute-to-CAP attribute comparisons of capabilities, such that CAP_{IC}s from different CAP_Os describing different users/systems/ATs/environments all use the same *Capability names*.

Where *Capability names* are the same, then the implication exists that the units of measure used for the *Capability values* are also the same. Algorithms comparing *Capability values* will need some awareness of units of measure to determine that the same unit is used. For example, if one CAP capability is entered in metres and another is in terms of nautical miles[2], then to allow comparison, one of the entries will need to be converted so both are in the same unit. This may be aided by the fact that *Capability values* includes a *unit* attribute.

Since the SI allows for twenty prefixes to form decimal multiples and submultiples of SI units (NIST, 1995), algorithms comparing *Capability values* will also need some ability to convert entries with different prefixes. For example, if one CAP capability is entered in terms of terahertz and another

[2]The "nautical mile" is a unit outside the SI that is currently accepted for use with the SI, subject to further review (1 nautical mile = 1852 m) (NIST, 1995).

is in terms of kilohertz, then to allow comparison, one of the entries will need to be converted so both are in the same unit and multiple.

6.3.2.2 Precision of CAP-Based Metrics

To be useful, metrics should display some degree of precision. Using a single metric is not allowed by the definition of accessibility used in this Thesis. This is primarily a result of the fact that accessibility is defined in terms of usability and usability is in turn defined in terms of three dimensions (i.e., effectiveness, efficiency, and satisfaction) each with a variety of attributes that potentially could be measured; thus, no one metric can be used to measure accessibility. However, there is a demand for greater precision in measuring accessibility, even though it really cannot be assessed in terms of one number, by definition.

There are two ways to achieve greater precision with the CAP. The first is to gather more possible values for a single attribute. The second is to use more attributes. The CAP actually does both:

a) There are several attributes for which the CAP documents multiple values to ensure the best precision possible. For example, when describing the language(s) supported by a specific IC, the CAP documents all of the languages used, whether one or many.

b) The primary way the CAP ensures greater precision by using more attributes is the design of its own structure. The CAP structure is designed to document several attributes across its entire specification. For example, the CAP notes the media, media types, languages, and other applicable capabilities to, as precisely as possible, document an IC.

Using multiple metrics to measure the concept of accessibility clearly has great potential and could meet the demand for greater precision.

Adding some new criteria to descriptors could also add greater precision. Such criteria could possibly be based on the absolute scale. However, such criteria can only do so if the users of the CAP are willing to accurately provide and use information with this level of precision. Currently placing such an expectation on the user is not justified.

Perhaps, as the CAP comes into wide-spread use, this additional precision could be added. However, since operators cannot compare unequal metrics, provisions will need to be made to deal with the differing amounts of precision presented when comparing the CAPs of users and systems. These provisions may require:

a) ignoring more precise CAP specifications and only dealing with specifications that are at the same level of precision – which may require converting specifications at "high level" precision to specific "low level" defaults,

b) (conversely) assuming defaults for missing precision – which is potentially very dangerous, or

c) requiring the gathering and input of further specifications to raise less precise specifications to the level of precision of the more precise ones – which then places greater demand on the user or system.

6.3.2.3 Use of Logical Operators

Drawing on two of the binary algorithms discussed in Chapter 4, AND and OR, the same kinds of logical operators might be used (with possible additional operators) when comparing CAP attribute-to-CAP attribute (and, by definition, over whole CAPs). Using these operations to combine CAPs raises an additional issue: whether accessibility can be purely defined in binary terms. In other words, is accessibility all or nothing?

On the one hand, it is clear that where enough limitations exist, no access can occur. For example, if the system keyboard is in a room with no windows and no artificial lighting (i.e., its pitch black), the user is not going to be able to see the keyboard. Furthermore, unless the user is a skilled touch typist or the keyboard is backlit, there is no way the user can effectively use the keyboard.

On the other hand, the situation may not be so black and white. For example, if the environment is significantly louder than the system's auditory output, the chances of a user, with "normal" hearing, hearing the auditory output may be very low, but there is still a possibility that the user may hear the output (although this does not mean they can hear enough to *understand* the auditory output). So it is possible to think, in the case of hearing, in non-binary terms. In essence, it may be more useful to know; for example, that the user can hear at 40 decibels and higher, the system is outputting auditory information at 60 decibels and the environment has background noise in the 60 to 80 decibel range, than to know if the user can or cannot hear the output, because the likelihood of the user hearing the output can be computed (however, the likelihood of the user being able to understand the output cannot).

Chapter 7

Discussion and Conclusion

This chapter reviews the problem investigated by this Thesis and discusses the potential for the CAP to resolve these issues. The chapter describes future directions of this research and potential future applications of the CAP. Finally, this chapter ends with the conclusion of this Thesis and a statement of its contribution.

7.1 Discussion

People have difficulties using computers; some people have more difficulties than others. These people can be of any age, ability, or background. To relieve these difficulties, computer users need standardized ways of identifying their access needs to systems and of identifying systems (and their components) that can meet their access needs.

While, it may not be possible to design a product that can be used effectively by all users in all contexts of use, accessibility is most readily provided by systems that are designed for the widest possible range of users and conditions. In addition to providing accessibility for trained/experienced users, it is also important to ensure accessibility for first time users. While system developers should not rely on user supplied ATs to achieve accessibility, in some cases, intermediaries such as ATs may still be required to achieve accessibility. These technologies transform interactions so that they can interact with different user/system capabilities.

Accessibility features, available in most operating systems, are useless if not accessible (International Organization for Standardization [ISO] and International Electrotechnical Commission [IEC], 2006b). They are generally unavailable until the user has successfully passed the log-in screen. However, without access to these features, many users may be unable to log-in. Thus, they need to be available *at* log-in, not after log-in.

"Closed systems", such as kiosks, present further problems, by disallowing user installation of AT software or hardware, in addition to any already supported, which could assist users in their log-in / use and setting of native accessibility features. Often, such systems limit the types of preference settings a user has access to.

To create a fully accessible log-in, the operating system might need to support an excessively multi-modal log-in interface for all users that would be a distraction to most users.

Among the guidelines of ISO 9241-171, the provision of Clause 8.3.5 recommends that operating systems provide and support user preference profiles:

> Software should enable users to create, save, edit and recall profiles of preferences, including input and output characteristics, without having to edit configuration files or carry out any restart that would cause a change of state or data (ISO, 2006f).

Using such profiles, a user can log-in to any computer on a network and expect certain settings and preferences to be maintained. But as currently implemented, user preference profiles do not provide an easy way for users to communicate their profile to other computers on other networks. Thus, this profile needs to cross network boundaries. Also, this profile needs to be acceptable to closed systems (Fourney & Carter, 2006a).

Users need to be involved in the selection, evaluation, and use of their assistive technologies. Research on AT use clearly demonstrates that consumers, who do not believe that they are involved in the selection of their AT(s), are more likely to discontinue use than those who feel involved (Riemer-Reiss & Wacker, 2000).

These difficulties suggest a need for an accessible (and portable) way to configure / activate accessibility features, and other user settings and preferences by a user upon arriving at a wide variety of computing devices. Instead of simply turning on/off specific technology features, which may or may not be present in a given system, there is a need to communicate the abilities of users to systems so that the system can self-adapt in the best possible way to meet the user's needs. A solution to these problems should focus on matching abilities, rather than having to make allowances for disabilities, and have a low possibility of abandonment by ensuring appropriate consumer input. The CAP meets these needs.

CAPs allow the consideration of multiple levels of system components, including: application software, operating systems, computer hardware, peripheral devices, and ATs. CAPs can be specified at various levels including:

- the overall combination of (interacting components) users, systems, ATs, and environments,
- individual interacting components (which can participate in various combinations at various times),
- individual inputs, outputs, and processes (component features) of interacting components, and
- individual details of the component features (that involve specific modalities, media, and other capabilities).

At the component feature level, information specific to an interacting component's modalities and any constraints placed on these media or processing of these media is recorded. Modality-

Specific information includes whether the media are visual, auditory, tactile, or olfactory in nature, what kind of media is involved (e.g., text, music), and the languages (and, where applicable, writing systems) supported. Capability-Specific information includes the capabilities of the media (e.g., in the case of sound, what pitches are available and what volume is supported). Processing-Specific information is used specifically to describe ATs because of their unique ability to change the modality (including supported languages) and/or capabilities of a medium. Processing-Specific information includes both the capabilities and kinds of transformation/translation of the information that occurs and whether or not the original information continues to be provided. This four level structure can identify sensory, physical, and linguistic accessibility issues and provides a basis for adding more detailed levels as they are needed. For example, the CAP could be extended to identify cognitive accessibility issues.

The CAP recognizes that accessibility is achieved and improved by serving "the widest variations within a context of use" (ISO, 2007k). This context of use should be based on the characteristics and variability of the user, the equipment, the environment, and the current task. The CAP is based on the UARM, illustrated in Figure 2.3 on page 19, which directly relates context(s) of use and accessibility. As noted in Chapter 2, the UARM focuses on the accessibility of interactions between systems and users that are working together in an environment to accomplish a set of tasks.

7.2 Contribution and Future Directions

The contribution of this Thesis is the development of a standardized way to describe the needs and capabilities of users and systems. This Thesis provides a framework that can be used on its own or in combination with other specifications such as ISO/IEC 24751.

A major internationally accepted result of the research discussed in this Thesis is ISO/IEC 24756. The evidence supporting this contribution and its importance is that this work was accepted in the CD and first FCD ballots of ISO/IEC 24756.

The following subsections describe a range of applications of the CAP.

7.2.1 Future Directions of Applying CAPs

CAP specifications have the potential to be easily combined and/or compared computationally. CAP specifications need to be combined when, for example, new components (e.g., programs or peripherals) are added to an existing computer system and the CAP of the new program or peripheral is added into the CAP of the computer system. CAP specifications are compared when users communicate their CAP to a system and the system then compares its own CAP with the user's CAP to establish an accessible interaction.

Once they can be compared, CAPs can be used to analyse existing human-computer interactions and to help evaluate the usefulness of proposed ATs. Comparing CAPs identifies "inaccessibilities" and can help users to select ATs that can reduce these inaccessibilities. A pair of CAPs can be used to compare the needs and abilities of systems with the abilities of users.

The following subsections describe potential applications that can be realized with on-going future research:

- Developing a User CAP
- Acquiring or developing system / environment / AT CAPs
- Using CAPs to customize a System
- Comparing CAPs to evaluate potential systems and/or ATs
- Using CAPs to specify legal and/or contractual requirements

Some of the requirements and issues for developing the CAP to achieve these applications are also explored.

7.2.2 Developing a User CAP

CAPs for users must be specifically individualized for each user. This requires that the CAP for a particular user be developed using an accessible tool that requires little or no configuration to support a wide range of users. The tool should work with a profile of various CAP possibilities to allow the user to select the most appropriate profile to describe his/her individual abilities and needs.

A basic interface could allow experienced users to directly specify their capability profiles. This interface could also be used by those who assist users with a disability to create profile(s) for these users.

A wizard-like interface could help novice users by asking them to select which of a number of CAP-related statements are true about them. For example, Jaycee, a nurse who is profoundly deaf in one ear, might use this wizard to establish her user CAP. She might start by selecting a statement like "I have trouble hearing". She would then see several statements describing various hearing states such as "I can only hear very loud sounds." and might select the statement "I can hear and understand someone whispering to me in a quiet room." as being descriptive of her abilities. The wizard would then present her with statements to discern her ability to hear binaurally like "I can hear stereo sound." where she might select "I can only hear monaural sound" (Fourney & Carter, 2006c). Because the CAP focuses on a person's abilities rather than their inabilities, positive statements (e.g., "I can see the colour red.") are preferred over negative statements ("I cannot see the colour red."). In addition to wizard-like interfaces that can support first-time and novice users, interfaces oriented to more experienced users and interfaces designed to assist users to further tweak their user CAP(s) may also be provided. In this manner, users would be able to easily use the CAP.

Tools used to support user CAP individualization must be as accessible as possible. Although many users may require assistance to use the tool to individualize a computing system for their use, the goal is to ensure that most users are able to use these tools independently of others. Independent use enables users to maintain the confidentiality of their personal information while further improving their access to the system.

A user's CAP can be stored locally on one's own system(s) as well as in a portable manner to use elsewhere. Copying a user's CAP to portable media means that a person's CAP(s) can be used at home and work as well as with publicly accessible systems. The user need only create their CAP once and copy it as needed, for example they may update specific copies for specific needs (e.g., home, work, public kiosks). Where publicly accessible services (i.e., so-called "closed systems") allow, the user can use the portable media carrying their CAP to communicate it to the service (see Section 7.2.4 for more information). This use of a portable CAP enables discrete, private, independent access to otherwise inaccessible computer systems.

7.2.3 Acquiring / Developing System / Environment CAPs

The international standardization of the CAP will encourage hardware and software manufacturers to create CAPs for their products. Manufacturers who claim conformance to the CAP standard, ISO/IEC 24756, would need to test their products to ensure full compliance. Any statement of compliance would be part of the advertised features of their product.

As more and more manufacturers provide CAPs, a publicly available database of CAPs for systems and their components could be developed. Such a database could even be updated by testing organizations, user organizations, and/or individual users. For example, the system CAP provided by a manufacturer could be further enhanced over time by adding user feedback. User feedback, such as consumer ratings and/or reviews, could differentiate solutions, assess their quality, and evaluate product appropriateness from a consumer perspective. User feedback, even if sparse or incomplete in nature, could identify the need for new products, as well as provide feedback to existing products.

Developing a CAP specification for a system or its components can be done in the same manner as already discussed for developing a user CAP. The need for testing to support certification of the validity of CAPs for their products suggests a role for people with disabilities to help manufacturers develop and validate CAPs for their products.

The development of CAPs for environments can only be performed once the characteristics of a particular environment are understood. At that point, the same tool used to develop user and system CAPs can be used. It is also possible that specialized programs that are environmentally aware (e.g., light sensors, microphone, etc.) could be used to automatically develop a CAP of the current environment. Since environmental conditions may change, the tools used to develop environmental CAPS must be easily accessible to modify these CAPs as required.

As with User CAP tools, tools to assist in the specification of a system or environment would need to be as accessible as possible supporting the widest range of users.

7.2.4 Using CAPs to Customize a System

All systems should have a base configuration that renders it accessible for the widest range of users, without the need for additional ATs or settings, in addition to enabling users to independently and efficiently communicate their user CAP to the system. Since a user's CAP can be potentially loaded onto a variety of different portable media, CAPs can be used both at home and work as well as with publicly accessible systems. One need only provide the public system with information about how best to interface with its user and any other systems (e.g., portable ATs) and their components being used within the current environment. A potential user should be able to just walk up and plug in.

The CAP does not specify the technology to use but current technology, such as Universal Serial Bus (USB) flash drives, small flash memory cards, or smart cards, could be used. There is also potential for technologies that interface with systems wirelessly such as PDAs, cellular telephones, or other devices that support Bluetooth (IEEE Computer Society, 2005). As the right technologies develop, it should be easy to use whatever technology a system supports.

Achieving such portability will require all systems to support a limited number of technologies so that users do not need to carry multiple portable devices with copies of their CAP(s). In addition, any devices designed to support portable CAPs should be a low-cost solution.

For example, Keiko, a student with low vision, needs to add more funds to her campus printing account using the campus services kiosk. This publicly available system has physically accessible and easy to tactilely find USB ports that support both the loading of user CAPs from portable media and the connection of USB-based portable AT(s). Keiko is easily able to find this USB port and plugs in a USB drive that contains her user CAP. A few seconds after Keiko's CAP is recognized and loaded into the system, the screen resets in high-contrast mode and the kiosk is ready to serve her. After she leaves, the system automatically reverts to its base configuration, to make it accessible for the maximum number of people without their needing to use a CAP of their own to change it (Fourney & Carter, 2006c).

This portability of a CAP potentially raises several issues of privacy and security. On the one hand, users need some sense that information provided to public systems remains confidential. On the other hand, users also require that information on portable devices remain secure. Complicating these issues is that owners of public systems require that portable CAPs not breach system security (if any), and that the means provided for CAP communication (whether wireless or not) not create opportunities for unintended uses that impact the public system's security or availability. Employers will have similar concerns with workplace systems.

7.2.5 Comparing CAPs to Evaluate Potential Systems / ATs

As mentioned in Chapter 4, typically systems are very inflexible and users are somewhat flexible. However, what should happen, if the system has alternatives available, but the user is very restricted in the number of alternatives they can use? Users need systems to have several alternative ways to access them. Using the CAP may help to identify the best available alternative(s). As the CAP analysis rules out each separate alternative, the list of available alternatives shrinks. At some point, there are no alternatives left, and it is necessary for the CAP to begin recommending the addition of AT(s).

At the same time, as mentioned above, users need to be involved in the selection, evaluation, and use of their ATs. Research on AT use clearly demonstrates that consumers, who do not believe that they are involved in the selection of their AT(s), are more likely to discontinue use than those who feel involved (Riemer-Reiss & Wacker, 2000).

However, many users are faced with barriers to accessing the information needed to fully participate in AT selection:

- Costs and attitudes mean that only a fraction of users with disabilities who use a computer also use an AT with their computer, even though many more report a need for an AT (Fichten et al., 2001).

- Counsellors, such as teachers and service providers, who help find computer-related ATs for use by people with disabilities admit not being very knowledgeable about them to the point of being unaware of ATs that may help (Bayha & Doe, 1998; Fichten et al., 2001).

- IT support personnel lack training in the use of ATs and consequently cannot assist users in the selection and maintenance of their ATs (Fichten et al., 2001).

- Consumers are in a catch-22, since information available regarding the ATs they need, even if in an electronic form, may not be accessible to them without an AT (Carter & Fourney, 2004a).

Thus, many users with disabilities have problems with finding, fitting, and supporting appropriate ATs to create an accessible interaction with systems. Using the CAP and its supporting tools, may make searching for appropriate AT fittings relatively easier. If and only if the CAP can access information, in the form of CAPs, about other available ATs, and then include this information into its evaluation for access, the CAP could support the user with finding, fitting, and supporting appropriate AT(s).

Suppose Nick, a legally blind software developer, wanted to improve his ability to access the various applications on his work computer. He would have to use his existing not entirely accessible system to search the web for other available options. He might also talk to friends about the ATs that they use, but these ATs might not actually help him. He may go to a local service agency for assistance in determining what further options he has, but he is restricted in what the counsellors at the agency know. Products that are new to the market and could help him might be missed in his search, because others have not heard of it. Nick may never find a particular product that could help him. Using his CAP along with a publicly available database of system (and system component) CAPs could provide considerable improvement over this scenario (Fourney & Carter, 2006c).

An existing system has already attempted to make the best possible match to Nick's needs based on his user CAP plus the system's own CAP and, where necessary, the CAP of its environment. It could then access a publicly available database and determine possible ATs that could improve accessibility. A publicly available database could similarly be used prior to purchasing new system components, or even complete new systems. This use could identify whether the proposed new configuration would be accessible, or if it would require one or more ATs to make it accessible.

For an AT to interface with the system, it needs to be compatible to the systems' properties (e.g. media, styles, operating systems and/or applications) as well as user capabilities (e.g., literacy). To interface successfully, the environment should not excessively handicap the accessibility of selected channels (e.g., noisy environments can handicap speech output). The addition of an AT may introduce new handicaps to interactions (e.g., the best choice may require a capability the user does not have) requiring additional ATs. Although AT fittings are not completely computable because each person is unique, using the CAP for this purpose may shorten the time and energy used to find appropriate AT(s).

A similar approach, using multiple different user CAPs, could be used to help select combinations of systems and ATs for use in multi-user settings, such as those provided by educational institutions.

Achieving this publicly available database raises several requirements and issues. Chief among them are issues related to security of the database and its data:

- User authentication and user tracking implemented by requiring individual users to log-in would allow every edit to be attributed to an individual user avoiding both inappropriate posting of content and unintentional deletions.

- Regular backup and recovery procedures to prevent loss of data in case of system failure.

- Posting usage guidelines that are short, simple and written in a positive tone would help users understand their roles (Powazek, 2002).
- A rollback feature could be used by administrators to repair any deletions or misuse as required.

The database would also need to be highly available ensuring 24/7/365 (24/7/366 for Leap Years) access. Users would be accessing this database at any time from any location via the Internet.

The community of users who access this database would likely be a mixture of users who use the CAP, users who want to support the CAP, users who are interested in what the CAP is all about, and users who want to crash or deface the database. There are three philosophical approaches to supporting this community:

First, a wiki could be created to provide publicly available CAP records. In a wiki approach, each CAP record (i.e., CAP_{IC}, CAP_{CF}, etc.) would be a wiki entry. Users would be encouraged to openly share the creation and editing of each CAP record. Users could add or alter any CAP record. Any change would be globally available immediately. Users would police changes to ensure consistent data. Users could download a CAP record right from the wiki to use in their own local CAP database. Users could directly upload new CAPs creating new wiki pages.

Second, an online content management system similar to Sourceforge could be created to provide publicly available CAP records. In this approach, each CAP record would be a project entry. Users could be asked to provide new records or edits to existing records, but no change would be made globally available, until the owner of the record approves the change. Using a Free/Libre/Open-Source Software (FLOSS) approach, the benefit of an open source community is maintained but certain individuals retain "ownership" over their CAP records and have the responsibility to ensure that their records remain correct and usable. Users could download a CAP record right from the site to use in their own local CAP database. Users could directly upload new CAPs creating new "projects" and having direct responsibility for them.

These first two approaches are essentially contributor-run digital libraries of CAPs. They allow people to contribute without major controls or gatekeeping by removing nearly all barriers to submission and instituting some simple verification procedures. Encouraging contributors to rank and comment on the contributions of others (even commenting on comments) could also add value and create a favourable environment for a "noisy, active, democratic community" to prosper (Jones, 2001). Digital libraries of this nature can also give back to contributors information, such as which CAPs are most in demand, in the form of top-ten lists or lists of most recommended CAPs (Jones, 2001).

A third, vastly different approach, arises since, with the creation of an International Standard based on the CAP, ISO could designate a centralized Registry Authority (RA) to manage publicly available CAP records. In the RA approach, the managers of the registry take sole responsibility for all records. Any new records or changes to existing records would need to be communicated to the RA for consideration and eventual inclusion. Any updates to existing records would be made available to the public very slowly. Users could still download a CAP record right from the site to use in their own local CAP database, but newer information would become available very slowly. This could potentially frustrate users, especially if detected errors are not corrected quickly enough.

7.2.6 Using CAPs to Specify Legal / Contractual Requirements

Legal accessibility requirements, such as Section 508 of the Rehabilitation Act (USA) (U.S. Federal Government, 1998), often focus on requiring specific system designs / components to meet the needs of specific sets of individual disabilities. Combining these requirements to create a system accessible to users with any of these individual disabilities, can create problems for both designers and users of systems, such as unnecessary limitations both in the design of systems and in the resulting usability of these systems (Fourney & Carter, 2006b). The resulting complexity of such an approach may limit the variety of users with disabilities, who are actually served by such accessibility requirements. For example, accessibility requirements tend to recognize the needs of the blind and the needs of the deaf as two distinct sets of needs, and to assume that the conjunction of these requirements will meet the needs of the deaf-blind. However, this is seldom the case.

Potentially, the CAP is the one current method of specifying *accessibility* needs that readily lends itself to conformance testing, because it is easy to compare the CAP of a system (or system component) to the CAP of a user. If legal / contractual expectations are developed in terms of CAPs, then there will be an even greater incentive for manufacturers to make their accessibility declaration, also in terms of CAPs (Fourney & Carter, 2006c).

Although they can help in the establishment of contracts or laws, International Standards are not developed as laws. However, legislation could be developed that specifies the CAPs of various sets of user needs that have to be met for a system to be declared "accessible". It is relatively easy to create a CAP for a deaf-blind user as well as CAPs for deaf or blind users (Fourney & Carter, 2006b).

While, the CAP may be considered as just another way of specifying *user* needs, it is potentially the one current method of specifying user needs that readily lends itself to conformance testing. By focusing on the abilities of the user, these CAPs provide a basis for assessing whether or not systems are accessible to them. By using a CAP, it may be unnecessary to further specify, in terms of accessibility requirements, how the system meets the user's needs (Fourney & Carter, 2006c).

7.2.7 Tools to Support these Applications

A future direction of this research is the analysis required to fully implement the CAP. This includes both determining the tools needed to support CAP management by users, as well as determining the best way to implement the files and databases necessary to support CAPs.

This research will involve determining the various tools required to fully support users of the CAP. Such tools would include:

- tools to support development and maintenance of User CAPs,
- tools to support manufacturer and community effort to develop and maintain System and AT CAPs,
- tools to support user-directed specification and maintenance of Environment CAPs, and
- tools to support sensor-directed specification and maintenance of Environment CAPs.

In addition, any further requirements necessary to develop software supporting the combination and comparison of CAPs to determine the needs of users and systems will also need to be understood.

Finally, the requirements needed for software that can apply the results of a CAP analysis to adapt the computing platform to fit the user also need to be understood.

The methodology required here is primarily one of analysis both in terms of user requirements and system requirements. The results of all previous research would be analysed to determine requirements and business rules needed for CAP implementation. This analysis will need to be detailed enough to allow development of an initial release of the CAP and an initial set of CAP tools for some computing platform (e.g., Windows, Mac, etc.).

Pursuing these future research directions allows the CAP to fulfil several potential applications.

7.3 Conclusion

Evaluating CAPs can determine the potential for the system and its components to meet the unique needs of the user (or groups of users with different needs). CAPs for systems, ATs, and their components provide a new format for presenting accessibility-related specifications. CAPs for users and environments could be easily developed using accessible tools.

The potential for a user's CAP to be stored onto a variety of different portable media, that are easily carried and can be used by a wide range of supporting devices could allow users with disabilities to discretely upload their individual needs into a system. The user can request that the system self-adapt in the best possible way to meet their needs. This self-adaptation could occur prior to user log-in and may even be acceptable to otherwise closed systems.

Using the CAP has great potential for getting users involved in the selection, evaluation and use of their AT(s), and may lead to their continued satisfactory use of the selected AT. This evaluative aspect of the CAP has potential for use in other areas, such as to determine the satisfaction of legal and/or contractual requirements.

Users with disabilities can use an individualized CAP to document specific system preferences. Specific user capabilities such as the restriction of low or high frequency sounds can be recorded. Specific environmental capabilities that directly affect the user's interaction with the system can be documented.

The CAP focuses on the CAPabilities of users, and systems and the minimizations of handiCAPs to interactions from the environment and incompatibilities among users, and systems.

This Thesis has described how the CAP provides a basis for identifying and dealing with accessibility issues in a standardized manner. CAPs can readily be developed for existing systems and their components, and can be custom developed for specific users and environments. It shows that CAPs can be applied to identifying handicaps to interactions between users and systems, and to evaluating and managing the use of systems and their components (including ATs added to systems), both for individual users and for select groups of specific users. CAPs can be used to help select and support combinations of systems and their components for use in multi-user settings. CAPs are also highly adaptable.

The development of ISO/IEC 24756, based on the CAP, will provide a commonly accepted method to evaluate a system's accessibility to a user. Further development of the CAP by ISO/IEC JTC1 SC35 will lead to widespread use of the CAP for these and other purposes.

"This is not the end. It is not even the beginning of the end.
But it is, perhaps, the end of the beginning."
Sir Winston Churchill

References

Ager, S. (1998). *Omniglot - writing systems and languages of the world.* (http://www.omniglot.com)

Akamatsu, M., & MacKenzie, I. S. (1996). Movement characteristics using a mouse with tactile and force feedback. *International Journal of Human-Computer Studies*, *45*(4), 483–493. (http://www.yorku.ca/mack/IJHCS2.html)

American National Standards Institute. (2001). *ANSI/NCITS 354: Common Industry Format for Usability Test Reports.*

Asakawa, C., Takagi, H., Ino, S., & Ifukube, T. (2003). Maximum listening speeds for the blind. In *Proceedings of the 2003 International Conference on Auditory Display (ICAD03), Boston, MA, USA.* (http://www.icad.org/websiteV2.0/Conferences/ICAD2003/paper/67%20Asakawa.pdf)

Baldwin, J. (2000). *The Baldwin Report.* (http://www.adcnetwork.net.au/BALDWIN%20REPORT.htm)

Basili, V., & Rombach, H. D. (1988). The TAME project: Towards improvement-oriented software environments. *IEEE Transactions on Software Engineering*, *14*(6), 758–773.

Bayha, B., & Doe, T. (1998). The internet: An inclusive magnet for teaching all students. In *Proceedings of 1998 CSUN Technology and Persons with Disabilities Conference, Los Angeles, CA.* (http://www.csun.edu/cod/conf/1998/proceedings/csun98_111.htm)

Bergman, E., & Johnson, E. (1995). Towards accessible human-computer interaction. In J. Nielsen (Ed.), *Advances in human-computer interaction* (Vol. 5, pp. 87–114). Norwood, New Jersey: Ablex Publishing Corporation. (http://www.sun.com/access/developers/updt.HCI.advance.html)

Bloom, B., et al. (1956). *Taxonomy of educational objectives: The classification of educational goals: Handbook I: Cognitive domain.* New York: Longmans, Green.

Byrum, J. D. (1999). ISO 639-1 and ISO 639-2: International standards for language codes. ISO 15924: International standard for names of scripts. In *Proceedings of the 65th International Federation of Library Associations and Institutions (IFLA) Council and General Conference, Bangkok, Thailand.* (http://www.ifla.org/IV/ifla65/papers/099-155e.htm)

Campanario, J. M. (1998). Peer review for journals as it stands today – Part 1. *Science Communication*, *19*(3), 181–211. (http://www2.uah.es/jmc/papers2.html)

Carter, J. (2004). *Putting usability first.* (Unpublished Manuscript)

Carter, J., & Fourney, D. (2004a). *The Common Accessibility Profile* (Tech. Rep. No. TR2004-02). Department of Computer Science, University of Saskatchewan. (http://www.cs.usask.ca/research/techreports/2004/TR-2004-02.pdf)

Carter, J., & Fourney, D. (2004b). Using a Universal Access Reference Model to identify further guidance that belongs in ISO 16071. *Universal Access in the Information Society*, *3*(1), 17–29. (http://www.springerlink.com/link.asp?id=wdqpdu5pj0kb4q6b)

Carter, J., Liu, J., Schneider, K., & Fourney, D. (2005). Transforming usability engineering requirements into software engineering specifications: From PUF to UML. In A. Seffah, J. Gulliksen, & M. Desmarais (Eds.), *Human-centered software engineering: Integrating usability in the development process* (pp. 145–167). Boston: Kluwer.

Charness, N., & Dijkstra, K. (1999). Age, luminance and print legibility in homes, offices and public places. *Human Factors*, *41*(2), 173–193.

Chin, J., Diehl, V., & Norman, K. (1988). Development of an instrument measuring user satisfaction of the human-computer interface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 213–218). ACM Press.

Chisholm, W., Vanderheiden, G., & Jacobs, I. (1999). *Web content accessibility guidelines 1.0* [W3C Recommendation]. World Wide Web Consortium. (http://www.w3.org/TR/WAI-WEBCONTENT/)

Clemens, S. (1889). *A Connecticut Yankee in King Arthur's court / by Mark Twain.* Toronto: G. M. Rose.

Colwell, C., Petrie, H., Kornbrot, D., Hardwick, A., & Furner, S. (1998). Haptic virtual reality for blind computer users. In *Proceedings of the Third International ACM Conference on Assistive Technologies* (pp. 92–99). ACM Press. (http://doi.acm.org/10.1145/274497.274515)

Constable, P., & Simons, G. (2002). An analysis of ISO 639: Preparing the way for advancements in language identification standards. In *Proceedings of the Twentieth International Unicode Conference (IUC20), Washington DC, USA.* (http://www.ethnologue.com/14/iso639/An_analysis_of_ISO_639.pdf)

Cooper, A. (1999). *The inmates are running the asylum.* Indianapolis, IN: Macmillan Computer Publishing.

Crane, R. T., & Hastings, B. L. (2001). *US Patent No. 6,664,948.* Washington, D.C.

Culler, D., Estrin, D., & Srivastava, M. (2004). Overview of sensor networks. *Computer*, *37*(8), 41–49. (http://www.computer.org/computer/homepage/0804/GEI/)

Dalby, D., & Gillam, L. (2004). "Weaving the linguasphere": LS 639, ISO 639 and ISO 12620. In *Proceedings of Workshop on A Registry of Linguistic Data Categories within an Integrated Language Resources Repository Area (INTERA) at LREC 2004, Lisbon.* (http://www.linguasphere.com/doc/lrec_workshop2004.pdf)

Davis, G. D., & Jones, R. (1989). *The sound reinforcement handbook* (2^{nd} ed.). Milwaukee, WI: Hal Leonard Corp.

Disabled Peoples' International. (1993). *Constitution of Disabled Peoples' International.* (http://www.dpi.org/en/about_us/constitution.htm)

D'Souza, M. E., & Lincoln, N. H. (2004). A persona-centric approach to developing complex computer systems: Lessons from the field. In *Proceedings of the Human Factors and Ergonomics Society (HFES) 48th Annual Meeting, New Orleans, LA* (pp. 917–921).

Edwards, A. (1995). Computers and people with disabilities. In A. Edwards (Ed.), *Extra-ordinary human-computer interaction: Interfaces for users with disabilities* (pp. 19–43). New York: Cambridge University Press.

Fagarasanu, M., & Kumar, S. (2003). Carpal tunnel syndrome due to keyboarding and mouse tasks: A review. *International Journal of Industrial Ergonomics*, *31*(2), 119–136. (http://dx.doi.org/10.1016/S0169-8141(02)00180-4)

Fay, R. R. (1988). *Hearing in vertebrates: A psychophysics databook.* Winnetka, IL: Hill-Fay Associates.

Fenton, N., & Pfleeger, S. (1997). *Software metrics: A rigorous and practical approach.* London: International Thomson Computer Press.

Fichten, C., Asuncion, J., & Barile, M. (2001). *Computer and information technologies: Resources for the postsecondary education of students with disabilities: Final report to the Office of Learning Technologies.* Hull, Québec: Office of Learning Technologies. (http://adaptech.dawsoncollege.qc.ca/pubs/olt01/)

Foraker Design. (2005). *Usability first glossary: cognitive load.* (http://www.usabilityfirst.com/glossary)

Fourney, D. (2004). *Information technology — Framework for establishing and evaluating accessibility in interactive systems ISO/JTC1/SC35/WG6 Document N0667.*

Fourney, D., & Carter, J. (2005). Initiating guidance on tactile and haptic interactions. In *Proceedings of the Guidelines On Tactile and Haptic Interactions (GOTHI-05) Conference, Saskatoon, Saskatchewan, Canada* (pp. 75–83). (http://userlab.usask.ca/GOTHI/Fourney%20-%20Guidance.pdf)

Fourney, D., & Carter, J. (2006a). Ergonomic accessibility standards. In *Proceedings of the 16th World Congress on Ergonomics (IEA 2006), Maastricht, Netherlands.*

Fourney, D., & Carter, J. (2006b). A standard method of profiling accessibility needs (using the CAP). In P. Isaias, M. McPherson, & F. Bannister (Eds.), *Proceedings of the IADIS International Conference e-Society 2006: Volume II* (pp. 138–142).

Fourney, D., & Carter, J. (2006c). A standard method of profiling the accessibility needs of computer users with vision and hearing impairments. In *Proceedings of the Conference on Assistive Technologies for Vision and Hearing Impairment (CVHI'2006), Kufstein, Austria.*

Garson, G. D. (n.d.). *PA 765 Statnotes: An online textbook.* (`http://www2.chass.ncsu.edu/garson/pa765/statnote.htm`)

Gibilisco, P. (2003). *People with different abilities: People with disabilities in the 21st century.* (`http://www.dsari.org.au/pgpaper.htm`)

Goodwin, D. (1997). *Underestimating the abilities of people with disabilities.* (`http://www.impactcil.org/Readings/understand.html`)

Groce, N. (1985). *Everyone here spoke sign language: Hereditary deafness on Martha's Vineyard.* Cambridge, MA: Harvard University Press.

Grudin, J., & Pruitt, J. (2002). Personas, participatory design and product development: An infrastructure for engagement. In *Proceedings of the Participatory Design Conference (PDC 2002), Malmo, Sweden* (pp. 144–161). (`http://research.microsoft.com/research/coet/Grudin/Personas/Grudin-Pruitt.pdf`)

Hanna, L., Risden, K., Czerwinski, M., & Alexander, K. (1998). The role of usability research in designing children's products. In A. Druin (Ed.), *The design of children's technology* (pp. 3–26). San Fransisco: Morgan Kaufman. (`http://www.research.microsoft.com/users/marycz/druin98.htm`)

Hjulstad, H. (2003). *ISO 639-6 (RE: Proposed Successor to RFC 3066 (language tags)).* (`http://eikenes.alvestrand.no/pipermail/ietf-languages/2003-November/001589.html`)

Hong Kong Equal Opportunities Commission. (1999, December). VII Summary of proceedings: Cluster three: Definitions of disability. In *Report of the Interregional Seminar and Symposium on International Norms and Standards relating to Disability.* Hong Kong, Special Administrative Region of China. (`http://www.worldenable.net/hongkong99/hkreport9907.htm`)

IBM Corporation. (2002). *IBM software accessibility checklist.* (`http://www-3.ibm.com/able/guidelines/software/accesssoftware.html`)

IEEE Computer Society. (2005). *IEEE Std 802.15.1-2005 (Revision of IEEE Std 802.15.1-2002) IEEE standard for information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements Part 15.1: Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs).* USA: Institute of Electrical and Electronics Engineers, Inc. (`http://standards.ieee.org/getieee802/download/802.15.1-2005.pdf`)

IMS Global Learning Consortium, Inc. (2003a). *IMS AccessForAll meta-data specification - Version 1.* (`http://www.imsglobal.org/accessibility/`)

IMS Global Learning Consortium, Inc. (2003b). *IMS Learner Information Package Accessibility for LIP - Version 1.* (`http://www.imsglobal.org/accessibility/`)

IMS Global Learning Consortium, Inc. (2003c). *IMS Learner Information Package Accessibility for LIP - Version 1 Final Specification - XML Example AccessForAll.* (`http://www.imsglobal.org/xsd/AccessForAll.xml`)

Inclusion of Disabled and Elderly People in Telematics. (2000). *Factors affecting uptake and use of technology.* (`http://www.stakes.fi/include/incc320.html`)

International Electrotechnical Commission. (1999). *IEC 61966-2-1: Multimedia systems and equipment - Colour measurement and management - Part 2-1: Colour management - Default RGB colour space - sRGB.* Geneva, Switzerland: International Electrotechnical Commission.

International Organization for Standardization. (1997a). *ISO 9241-14:1997: Ergonomic requirements for office work with visual display terminals (VDTs) — Part 14: Menu dialogues.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization. (1997b). *ISO 9241-15:1997: Ergonomic requirements for office work with visual display terminals (VDTs) — Part 15: Command dialogues.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization. (1998a). *ISO 9241-11:1998: Ergonomic requirements for office work with visual display terminals (VDTs) — Part 11: Guidance on usability.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization. (1998b). *ISO 9241-17:1998: Ergonomic requirements for office work with visual display terminals (VDTs) — Part 17: Form filling dialogues.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization. (1999a). *ISO 13407:1999: Human-centred design processes for interactive systems.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization. (1999b). *ISO 9241-16:1999: Ergonomic requirements for office work with visual display terminals (VDTs) — Part 16: Direct manipulation.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization. (2002a). *ISO 14915-3:2002 Software ergonomics for multimedia user interfaces — Part 3: Media selection and combination.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization. (2002b). *ISO 639-2:2002 Codes for the representation of names of languages — Part 2: Alpha-3 code.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization. (2003a). *ISO 14915-2:2003 Software ergonomics for multimedia user interfaces — Part 2: Multimedia control and navigation.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization. (2003b). *ISO/TS 16071:2003 Ergonomics of human system interaction — Guidance on accessibility for human computer interfaces.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization. (2004a). *ISO 15924:2004 Information and documentation – Codes for the representation of names of scripts.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization. (2004b). *ISO 8601:2004 Data elements and interchange formats – Information interchange – Representation of dates and times.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization. (2006a). *ISO 80000-3:2006 Quantities and units – Part 3: Space and time.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization. (2006b). *ISO 80000-4:2006 Quantities and units – Part 4: Mechanics.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization. (2006c). *ISO CD 80000-10:2006 Quantities and units – Part 10: Atomic and nuclear physics.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization. (2006d). *ISO CD 80000-1:2006 Quantities and units – Part 1: General.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization. (2006e). *ISO CD 80000-2:2006 Quantities and units – Part 2: Mathematical signs and symbols to be used in the natural sciences and technology.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization. (2006f). *ISO DIS 9241-171:2006 Ergonomics of human system interaction — Guidance on accessibility for human computer interfaces.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization. (2007a). *ISO 639-3:2007 Codes for the representation of names of languages — Part 3: Alpha-3 code for comprehensive coverage of languages.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization. (2007b). *ISO 80000-5:2007 Quantities and units – Part 5: Thermodynamics.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization. (2007c). *ISO 80000-8:2007 Quantities and units – Part 8: Acoustics.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization. (2007d). *ISO DIS 80000-11:2007 Quantities and units – Part 11: Characteristic numbers.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization. (2007e). *ISO DIS 80000-12:2007 Quantities and units – Part 12: Solid state physics.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization. (2007f). *ISO DIS 80000-7:2007 Quantities and units – Part 7: Light.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization. (2007g). *ISO DIS 80000-9:2007 Quantities and units – Part 9: Physical chemistry and molecular physics.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization. (2007h). *ISO FDIS 80000-13:2007 Quantities and units – Part 13: Information science and technology.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization. (2007i). *ISO FDIS 80000-14:2007 Quantities and units – Part 14: Telebiometry related to human physiology.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization. (2007j). *ISO FDIS 80000-6:2007 Quantities and units – Part 6: Electromagnetism.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization. (2007k). *ISO FDIS 9241-20:2007: Ergonomics of human-system interaction — Part 20: Accessibility guidelines for information/communication technology (ICT) equipment and services.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization, & International Electrotechnical Commission. (1998a). *ISO/IEC 14598-1:1998 Information technology — Software product evaluation — Part 1: General overview.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization, & International Electrotechnical Commission. (1998b). *ISO/IEC 8859-1:1998 Information technology — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization, & International Electrotechnical Commission. (2001a). *ISO/IEC 9126-1:2001 Information technology — Software product quality — Part 1: Quality model.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization, & International Electrotechnical Commission. (2001b). *ISO/IEC Guide 71:2001 Guidelines for standards developers to address the needs of older persons and persons with disabilities.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization, & International Electrotechnical Commission. (2002). *ISO/IEC 9995-3:2002 Information technology — Keyboard layouts for text and office systems — Part 3: Complementary layouts of the alphanumeric zone of the alphanumeric section.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization, & International Electrotechnical Commission. (2004). *ISO/IEC Directives, Part 1* (5 ed.). Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization, & International Electrotechnical Commission. (2006a). *ISO/IEC 25062:2006 Software engineering – Software product quality requirements and evaluation (SQuaRE) – Common industry format (CIF) for usability test reports.* Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization, & International Electrotechnical Commission. (2006b). *ISO/IEC CD 24786:2006 Information technology – User interfaces – Accessible user interface for accessibility setting on information devices Part 1: General (ISO/IEC JTC 1 / SC35 Document N1025).* Geneva, Switzerland: International Organization for Standard-

ization.
International Organization for Standardization, & International Electrotechnical Commission. (2006c). *ISO/IEC FCD 24751-2:2006 Individualized adaptability and accessibility in e-learning, education and training – Part 2: Access for all personal needs and preferences statement.* Geneva, Switzerland: International Organization for Standardization.
International Organization for Standardization, & International Electrotechnical Commission. (2006d). *ISO/IEC FCD 24751-3:2006 Individualized adaptability and accessibility in e-learning, education and training – Part 3: Access for all digital resource description.* Geneva, Switzerland: International Organization for Standardization.
International Organization for Standardization, & International Electrotechnical Commission. (2006e). *ISO/IEC FDIS 24751-1:2006 Individualized adaptability and accessibility in e-learning, education and training – Part 1: Framework.* Geneva, Switzerland: International Organization for Standardization.
International Organization for Standardization, & International Electrotechnical Commission. (2006f). *ISO/IEC JTC1/SC7 N3425 new work item proposal: System and software product quality requirements and evaluation (SQuARE) – Common industry format for usability – General framework for usability-related information.* Geneva, Switzerland: International Organization for Standardization.
International Organization for Standardization, & International Electrotechnical Commission. (2006g). *ISO/IEC JTC1/SC7 N3431 letter ballot on the establishment of a JWG between ISO TC 159/SC4 and JTC1/SC7 (administered by SC7) on the Common industry Format for usability.* Geneva, Switzerland: International Organization for Standardization.
International Organization for Standardization, & International Electrotechnical Commission. (2007a). *ISO/IEC FCD 24756:2007 Information technology – Framework for specifying a common access profile (CAP) of needs and capabilities of users, systems, and their environments.* Geneva, Switzerland: International Organization for Standardization.
International Organization for Standardization, & International Electrotechnical Commission. (2007b). *ISO/IEC PDTR 29128-1:2007 Information technology — Accessibility considerations for people with disabilities — Part 1: User needs summary.* Geneva, Switzerland: International Organization for Standardization.
International Organization for Standardization, & International Electrotechnical Commission. (2007c). *ISO/IEC PDTR 29128-2:2007 Information technology — Accessibility considerations for people with disabilities — Part 2: Standards inventory.* Geneva, Switzerland: International Organization for Standardization.
International Organization for Standardization, & International Electrotechnical Commission. (2007d). *ISO/IEC PDTR 29128-3:2007 Information technology — Accessibility considerations for people with disabilities — Part 3: Guidance on user needs mapping.* Geneva, Switzerland: International Organization for Standardization.
International Organization for Standardization, & International Electrotechnical Commission SC7 Study Group. (2005). *ISO/IEC JTC1/SC7 N3189 SC7 Common Industry Format (CIF) study group report.* Geneva, Switzerland: International Organization for Standardization.
Irish National Body, Deaf Action Committee for SignWriting. (1999). *Proposed ISO 639-2 code for sign languages.* (http://www.evertype.com/standards/iso639/sign-language.html)
ISO JTC 1 SWG Accessibility. (2006). *User needs summary version 1.0 (ISO JTC1 SWG-A N212).* (http://www.jtc1access.org/documents/swga_212.zip)
Jacko, J., & Vitense, H. (2001). A review and reappraisal of information technologies within a conceptual framework for individuals with disabilities. *Universal Access in the Information Society, 1*(1), 56–76. (http://springerlink.com/link.asp?id=yrnp6lvy63ge7drd)
Jones, P. (2001). Open (source)ing the doors for contributor-run digital libraries. *Communications of the ACM, 44*(5), 45–46. (http://doi.acm.org/10.1145/374308.374337)
Kaplan, D. (n.d.). *The definition of disability.* (http://www.accessiblesociety.org/topics/demographics-identity/dkaplanpaper.htm)
Kaye, J. N. (2001). *Symbolic olfactory display.* Unpublished master's thesis, Massachusetts Institute of Technology. (http://alumni.media.mit.edu/~jofish/thesis/)

Kotzé, P., Eloff, M. M., Adesina-Ojo, A., & Eloff, J. H. P. (2004). Accessible computer interaction for people with disabilities: The case of quadriplegics. In *Proceedings of the 6th International Conference on Enterprise Information Systems (ICEIS 2004): Volume 5* (pp. 97–106). (http://www-lih.univ-lehavre.fr/Intranet/proceedings/ICEIS2004/ICEIS%202004/Area%205%20-%20Human-Computer%20Interaction/ICEIS%202004%20Volume%205%20-%20HCI.pdf)

Lynch, G., & Meads, J. (1986). In search of a user interface reference model. *SIGCHI Bulletin*, *18*(2), 25–33. (http://doi.acm.org/10.1145/15683.15686)

Mackintosh, N. (1998). *IQ and human intelligence*. New York: Oxford University Press.

MacLennan, B. J. (1983). *Principles of programming languages: Design, evaluation, and implementation*. New York: Holt, Rinehart, and Winston.

Macleod, M. (1994). Usability in context: Improving quality of use. In *Fourth IEA International Symposium on Human Factors in Organizational Design and Management*. Stockholm, Sweden. (http://www.usability.serco.com/papers/mm-con94.pdf)

Manlove, E. E., Frank, T., & Vernon-Feagans, L. (2001). Why should we care about noise in classrooms and child care settings? *Child and Youth Care Forum*, *30*(1), 55–64. (http://dx.doi.org/10.1023/A:1016663520205)

Maskery, H., & Meads, J. (1992). Context: In the eyes of users and in computer systems. *SIGCHI Bulletin*, *24*(2), 12–21. (http://doi.acm.org/10.1145/142386.142389)

McQuaid, P. A. (1996). *Profiling software complexity*. Unpublished doctoral dissertation, Auburn University.

Messick, S. (1995). Validity of psychological assessment: Validation of inferences from persons' responses and performance as scientific inquiry into scoring meaning. *American Psychologist*, *50*(9), 741–749.

Microsoft Corporation. (2000). *Microsoft Windows 2000 Professional Resource Kit*. Redmond, Washington: Microsoft Press. (http://www.microsoft.com/resources/documentation/windows/2000/professional/reskit/en-us/default.mspx)

Moulton, G., Huyler, L., Hertz, J., & Levenson, M. (2002). *Accessible technology in today's business: Case studies for success*. Redmond, Washington: Microsoft Press.

Munson, J. (2003). *Software engineering measurement*. New York: Auerbach Publications.

Nadol, J. B. (1993). Hearing loss. *New England Journal of Medicine*, *329*(15), 1092–1102.

Nielsen, J. (2002). *Alert box: Usability for senior citizens*. (http://www.useit.com/alertbox/20020428.html)

O'Connor, T. (2004). *Measurement in research*. (http://faculty.ncwc.edu/TOConnor/308/308lect04.htm)

Origin Instruments Corporation. (2007). *Origin Instruments Corporation Homepage*. (http://www.orin.com)

Pandian, C. R. (2004). *Software metrics: A guide to planning, analysis, and application*. New York: Auerbach Publications.

Powazek, D. M. (2002). *Design for community: The art of connecting real people in virtual places*. Indianapolis, IN: New Riders.

Raggett, D., Hors, A. L., & Jacobs, I. (1999). *HTML 4.01 Specification* [W3C Recommendation]. World Wide Web Consortium. (http://www.w3.org/TR/html401)

Reed, C., & Farah, M. (1995). The psychological reality of the body schema: A test with normal participants. *Journal of Experimental Psychology: Human Perception & Performance*, *21*(2), 334–343.

Riemer-Reiss, M. L., & Wacker, R. R. (2000). Factors associated with assistive technology discontinuance among individuals with disabilities. *Journal of Rehabilitation*, *66*(3), 44–50. (http://www.findarticles.com/p/articles/mi_m0825/is_3_66/ai_66032259)

Rioux, M. (1997, April). Disability: The place of judgment in a world of fact. *Journal of Intellectual Disability Research*, *41*(2), 102–111.

Rossotti, H. (1983). *Color*. Princeton, NJ: Princeton University Press.

Shannon, C. (1948, July and October). A mathematical theory of communication. *Bell System Technical Journal*, *27*, 379–423 and 623–656.

Shannon, C., & Weaver, W. (1949). *The mathematical theory of communication.* Urbana, Illinois: University of Illinois Press.

Shardanand, U., & Maes, P. (1995). Social information filtering: algorithms for automating "word of mouth". In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 210–217). (http://doi.acm.org/10.1145/223904.223931)

SIL International. (2006). *Ethnologue, languages of the world.* (http://www.ethnologue.com)

Simons, G. F. (2000). Language identification in metadata descriptions of language archive holdings. In *Proceedings of Workshop on Web-Based Language Documentation and Description.* (http://www.ldc.upenn.edu/exploration/expl2000/papers/simons/simons.htm)

Sloan, D., Gregor, P., Rowan, M., & Booth, P. (2000). Accessible accessibility. In *Proceedings of the 2000 Conference on Universal Usability* (pp. 96–101). ACM Press. (http://doi.acm.org/10.1145/355460.355480)

Stephanidis, C., & Savidis, A. (2001). Universal access in the information society: Methods, tools, and interaction technologies. *Universal Access in the Information Society*, *1*(1), 40–55. (http://springerlink.com/link.asp?id=gbxy7hthk45hu3q2)

Stevens, S. (1959). Measurement, psychophysics, and utility. In C. Churchman & P. Ratoosh (Eds.), *Measurement: Definitions and theories* (pp. 18–63). New York: Wiley.

Stokes, M., Anderson, M., Chandrasekar, S., & Motta, R. (1996). *A standard default color space for the internet, version 1.10.* (http://www.w3.org/Graphics/Color/sRGB)

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, *12*(1), 257–285.

Taubes, G. (2000). Transparent computing. *Think Research*, *6*(1). (http://domino.watson.ibm.com/comm/wwwr_thinkresearch.nsf/pages/coverstory100.html)

Thatcher, J., Waddell, C., Henry, S., Swierenga, S., Urban, M., Burks, M., et al. (2002). *Constructing accessibile web sites.* Birmingham, U.K.: Glasshaus.

Thimbleby, H. (1995). Treat people like computers? Designing usable systems for special people. In A. Edwards (Ed.), *Extra-ordinary human-computer interaction: Interfaces for users with disabilities* (pp. 283–295). New York: Cambridge University Press.

Tohma, Y. (2004). Incorporating fault tolerance into an autonomic-computing environment. *IEEE Distributed Systems Online*, *5*(2). (http://csdl.computer.org/comp/mags/ds/2004/02/o2003.pdf)

United Nations Commission for Social Development on Disability. (1993). *United Nations standard rules on the equalization of opportunities for persons with disabilities (A/RES/48/96).* (http://www.independentliving.org/standardrules/)

United States Department of Commerce National Institute of Standards and Technology [NIST]. (1995). *Guide for the Use of the International System of Units (SI) by Barry N. Taylor.* U.S. Government Printing Office (NIST Special Publication 811). (http://physics.nist.gov/Document/sp811.pdf)

U.S. Federal Government. (1998). *Section 508 of the Rehabilitation Act.* (http://www.section508.gov/)

Vanderheiden, G. (1990). Thirty-something million: Should they be exceptions? *Human Factors*, *32*(4), 383–396. (http://trace.wisc.edu/docs/30_some/30_some.htm)

Vanderheiden, G. (2007, November 29). Personal Communication.

Vitense, H. S., Jacko, J. A., & Emery, V. K. (2002). Foundation for improved interaction by individuals with visual impairments through multimodal feedback. *Universal Access in the Information Society*, *2*(1), 76–87. (http://www.springerlink.com/content/pj6y9yu2uum9qhy0)

Weaver, W. (1949). Recent contributions to the mathematical theory of communication. In C. Shannon & W. Weaver (Eds.), *The mathematical theory of communication* (pp. 93–117). Urbana, Illinois: University of Illinois Press.

West, T. (1999). The abilities of those with reading disabilities: Focusing on the talents of people with dyslexia. In D. Duane (Ed.), *Reading and attention disorders: Neurobiological correlates* (pp. 213–241). Baltimore, Maryland: York Press.

Wikipedia. (2007). *Police 911 — Wikipedia, the free encyclopedia.* (http://en.wikipedia.org/w/index.php?title=Police_911&oldid=164615936)

Winter, D. T. (n.d.). *Scripts for the blind.* (http://homepages.cwi.nl/~dik/english/codes/braille.html)

World Health Organization. (2002). *Towards a common language for functioning, disability and health: ICF*. Geneva, Switzerland: World Health Organization. (http://www3.who.int/icf/beginners/bg.pdf)

Zimmerman, T. (1996). Personal Area Networks: Near-field intrabody communication. *IBM Systems Journal*, *35*(3 and 4), 609–617. (http://researchweb.watson.ibm.com/journal/sj/353/sectione/zimmerman.html)

Appendix A

Glossary

This Appendix contains definitions of the terms used specifically for this Thesis.

Common Accessibility Profile (CAP) A user-system interaction model based on the UARM that describes the set of potential interactions between ICs.

CAP(Assistive Technology) (CAP_{AT}) A CAP used to describe an AT interacting component.

CAP(Capabilities) (CAP_{C}) The level of the CAP which documents any capabilities of the IC (e.g., SI units, capabilities from other standards) whether or not they are used in the interaction.

CAP(Component Features) (CAP_{CF}) The actual level of the CAP responsible for interactions between ICs. This is the level for documenting inputs (IRs), outputs (OTs), and any processing (PFs) unique to ATs.

CAP(Environment) (CAP_{ENV}) A CAP used to describe an Environment interacting component.

CAP(Input Receptor) (CAP_{IR}) A CAP used to describe an input CF (i.e., an IR).

CAP(Interacting Component) (CAP_{IC}) A CAP used to describe an interacting component. This CAP can be of type CAP_{USE}, CAP_{SYS}, CAP_{AT}, or CAP_{ENV}.

CAP(Modality) (CAP_{M}) The level of the CAP which documents the properties of the interaction. This includes the languages, media, and modalities (visual, auditory, tactile, olfactory) used.

CAP(Output Transmitter) (CAP_{OT}) A CAP used to describe an output CF (i.e., an OT).

CAP(Overall) (CAP_{O}) A high level container for all CAPs.

CAP(Processing) (CAP_{P}) The level of the CAP which documents any transformations of the media or language that occurs in the interaction. These transformations are usually created by ATs.

CAP(Processing Function) (CAP_{PF}) A CAP used to describe a processing CF (i.e., a PF).

CAP(System) (CAP_{SYS}) A CAP used to describe a System interacting component.

CAP(User) (CAP_{USE}) A CAP used to describe a User interacting component.

Component Feature (CF) The features and properties of an IC. In the UARM, these features and properties include media, languages, and interaction styles. Originally known as "Interacting Component Feature", this term is shortened to "Component Feature" to avoid confusion with the term IC.

Handicap Anything that may interfere with the accessibility of interactions. A handicap is not a condition of the user.

Input Receptor (IR) A CF which receives communications. Communication is only possible where there are corresponding OTs for the IRs being used. According to the UARM, IRs can be specified in terms of interacting component, direction, modality, and properties.

Interaction A point of communication between two ICs. In the UARM and CAP models, interactions occur across channels in one direction only.

Interacting Component (IC) The elements of the CAP which interact with each other. Systems, Users, ATs, and Environments are considered to be interacting components.

Output Transmitter (OT) A CF which transmits communications. Communication is only possible where there are corresponding IRs for the OTs being used. According to the UARM, OTs can be specified in terms of interacting component, direction, modality, and properties.

Processing Function (PF) The logical portion of an AT which may transform communications between inputs and outputs. For the system, a PF is the application logic interacting with the data store, and making use of interaction components to produce the systems interface. For a user, a PF is equivalent to one's cognitive capabilities. According to the UARM, processing can be specified in terms of IR, transformation, and OT.

Universal Access Reference Model (UARM) A user-system interaction model based on a human factors perspective and related international standards. The UARM suggests that the total set of interactions makes up the potential user experience of the system.

Appendix B

Example of Gesture in Interaction

In Section 2.4.1.4, gesture is discussed for its potential in innovative interactive designs. As an example, the Konami arcade game Police 911 (also marketed as Police 24/7 in Europe and The Keisatsukan Shinjuku 24ji in Japan) is mentioned. This appendix provides additional information on this arcade game. For more information see Wikipedia (2007).

B.1 History

Released by Konami in 2000, Police 911 is a light gun arcade game that is 1,206 mm x 1,971 mm x 2,178 mm in size (see Figure B.1). The player is cast as either a "one man SWAT team" working for the Tokyo police, or an American police officer of the LAPD, working to take down members of the Gokudo-kai, an internationally-based yakuza group.

B.2 Gameplay

The game is highly interactive. Where most light gun games requires the player to stand in one place and shoot the enemies before being shot, this game uses a foot pad and infrared sensors to determine the player's location. This approach allows the simulation of the player being able to dodge (with the knees, while standing on the pad), duck to avoid bullets (and reload), and lean out to maximise cover and get a better shot. Since enemies will continue to shoot, while the player is "hiding", it is possible in this game to "rise up" and immediately get shot. In addition, the game has a time limit such that the player cannot hide for very long.

Figure B.1: The arcade game "Police 911"

Appendix C

Notes on CAP Implementation

This Appendix considers ideas and issues that may prove useful when implementing the CAP.

C.1 Data Format

Although the description of the CAP in this Thesis is implementation-independent, there are certain additional features that must be present in all CAPs. These features include the language of the CAP, the method of encoding (i.e., the character set), and information clearly labelling the content as a CAP specification.

For example, if a CAP is specified using XML (see Listing C.1), then it would begin with a header that defines the character encoding of the specification (default is ISO/IEC 8859-1 (International Organization for Standardization [ISO] and International Electrotechnical Commission [IEC], 1998b)) and a namespace declaration showing the specification type (i.e., CAP), the language of the specification (e.g., English, en), and the version of the specification (i.e., 1.0).

Listing C.1: Example XML Declaration for a CAP

```
<?xml version=``1.1'' encoding=``ISO-8859-1''?>
      <cap xmlns='http://www.iso.org/CAP/en/1.0' ...>
...
```

On the other hand, a specification based on SQL might use a different approach to specify this same information.

C.2 Data Security

To ensure performance and ease of use, it is not currently envisioned for CAPs, in particular CAP_{USE}s, to be encrypted. For this reason, highly sensitive personal information (e.g., Social Insurance Numbers, Health Insurance numbers, Credit Card numbers, etc.) should never be kept within a CAP.

If CAPs were encrypted, there would be several implications. First, encryption would affect how users approach the use of the CAP: users may be lulled into a sense of complacency as to the security of their personal data. Second, the act of encryption does not actually guarantee security — the encryption approach could at some point in the future be broken revealing the user's personal data.

Given CAPs may involve one or more ICs containing one or more sets of specifications of IRs, PFs, and OTs, CAPs have the potential to become very large. This suggests potential performance implications for an encrypted CAP. For example, users might experience unacceptable delay because of the need to decrypt the entire CAP across each system access. A system at the workplace may be perceived as responding differently than a public kiosk or a system located at home solely based on the need for decryption upon CAP access.

Finally, the use of encryption could unacceptably affect system accessibility. The use of encryption implies that the user and system would each need to provide some method of authentication, prior to the system being able to access the user's CAP and adapt accordingly. While this might be trivial for a system, it could prove to be impossible for a user.

Although CAP_{USE}s should not be encrypted, for the purpose of data security, all others probably can. For example, encrypting CAP_{SYS}s has the potential of protecting the system's settings, etc. from being modified by a malicious user or accessed by an unauthorized user with the explicit purpose of determining the system's weaknesses.

C.3 A Secure CAP?

It would be useful for CAPs to carry information such as usernames and other authenticating information so that when a user approaches a new system, the changes required by their CAP is used only if the user is authorized to use this system. For example:

- All Automated Teller Machines might accept all CAP_{USE}s for kiosk access, but only specific CAP_{USE}s for system access.
- A system in a private computer lab might only allow access by a specific group of CAP_{USE}s.
- A user's laptop might be set to only allow that user's CAP_{USE}.

This suggests the future need for a new type of CAP, specifically a "Secure CAP". Unlike a CAP_{USE}, a Secure CAP would be encrypted and only be used to allow an accessible means of authentication. Thus, a CAP_{USE} would be used to ensure access to the system is available, while the Secure CAP could then allow login.

It is also possible that existing CAP specifications could be expended in the future to carry several network profiles, user names, security keys, etc. instead of using a specialized CAP for this purpose. This would further suggest the need to only encrypt part of a CAP rather than the whole thing, or even the need for encryption of entries rather than encrypting the whole CAP.

However, this is beyond the scope of this Thesis.

Appendix D

Additional Attributes Beyond those Needed to Compute CAPs

This Appendix discusses how various CAP(s) could be expanded to include optional Type-Specific Information that could provide supplementary knowledge beyond that required to identify access issues.

D.1 $\mathbf{CAP}_{USE}$ Type-Specific Information

CAP_{USE} specifications could be expanded in the future to include CAP_{USE} Type-Specific Information such as personal or identifying information. For example, optional personal information, such as date of birth, could help the system adapt to the youth/maturity of the user.

Care should be taken when this kind of optional Type-Specific Information is requested for a CAP_{USE}. It is usually most desirable to use information that the user may not need to update in the future. For example, an attribute such as "age" would need to be updated at least annually by the user and would require either the user to remember to do so or the system to remind the user. On the other hand, "date of birth" need only be entered once (e.g., in the Calendar date format specified in ISO 8601 (see International Organization for Standardization [ISO], 2004b)) and then "age", if this information is needed, can be calculated from it.

D.2 $\mathbf{CAP}_{SYS}$ Type-Specific Information

It is envisioned that CAP_{SYS} specifications could be expanded in the future to include CAP_{SYS} Type-Specific Information such as product support, warranty, CAP versioning, and system dependencies.

Product support information could include technical support options and information describing the manufacturer. It is envisioned that information on how to reach technical support, either for this system and/or for accessibility of this system, would be provided including all possible reasonable contact information. For example:

- In-house/local technical support could be specified with a room number or address and local telephone number.

- Manufacturer-supplied technical support could be specified with one or more toll-free telephone numbers and/or electronic contact points (e.g., one or more web, e-mail, or instant messaging (IM) addresses) with which to electronically contact technical support.

Users will expect that at least one method would provide direct real-time access to technical support staff. Note ISO 9241-171 expects, in the case of telephone numbers, support for text telephones (TTY) to be noted and provided (ISO, 2006f).

Technical support information should contain the name of the person, business, or office being contacted. Including an address could help distinguish between local, outside, or manufacturer support. These types of information would allow a CAP_{SYS} to describe both manufacturer and local support options.

Contact information relevant to the type of technical support could also be documented. Given that telephone-based support could be by fax, phone, or TTY, this information could contain a fully-qualified telephone number including country, area, and city codes, as well as the type of support provided at this number. For support via the web, information such as the fully-qualified

URL and the full e-mail address of the support service should be provided. For support via IM, the information provided should include both the IM network supported (e.g., ICQ, AIM, MSN, Skype, etc.) and the support address (e.g., ICQ number, Skype username). For support via postal mail, information such as the valid postal mail address, including country and postal/zip code, for the support service could be provided. For "walk-in" support, a valid physical street address where users can access support could be provided.

In addition, CAP_{SYS} could also be expanded in the future to contain product information such as the serial, model, and/or version number of the system to help further identify the system and its components or assist with warranty information and CAP versioning.

Finally, in the future, IC dependencies might also be described through a CAP_{SYS}. For example, since some systems are dependant on the presence of other systems, these dependencies could be recorded in a CAP_{SYS}.

D.3 CAP_{ENV} Type-Specific Information

A CAP_{ENV} could be further specified using optional CAP_{IC} Type-Specific Information relevant to the environment. For example, in the case of mobile applications, it may be useful to know of any neighbouring environments. Thus, in a mobile system, a *Neighbours* attribute could be used to document the unique CAP *Name*s of neighbouring environments.

This attribute might later become computable such that the unique *Name* of the current environment might be used to compute or otherwise determine the *Name*s of neighbouring environments. The results of such a computation could then be kept in the *Neighbours* attribute.

In either case, such a feature would allow the CAP of a mobile system to be aware in advance of any needed changes, should the user move from one room to another.

Appendix E

ISO/IEC FCD 24756

This Appendix provides more information on ISO/IEC 24756, which is based on the work of this Thesis.

E.1 Notes

Currently, there are two differences between the CAP specifications of ISO 24756 and this Thesis.

1. While this Thesis refers to a CAP as a "Common Accessibility Profile", ISO 24756 refers to a "Common Access Profile".

2. In this Thesis, the use of channels by CFs is documented according to the attributes: *Capacity*, *Parallel*, *Serial*, *Degree*, *Intermittent*, and *Devoted*. This approach allows a CAP to document that a CF uses multiple channels in different ways.

   In ISO 24756, the use of channels by CFs is documented according to the attributes: *Channel capacity*, *Sharing capability*, *CF operations*, and *Priority*. This approach also assumes all channels are used by the CF in the same manner.

   - *Channel capacity* indicates the number of channels that a CF can be connected to at one time.
   - *Sharing capacity* indicates whether or not the CF can share a channel with other similar CFs. A sharing capacity can be "SHARABLE", "DEDICATED" (not sharable), or "POSSIBLE".
   - *CF operations* is one of INTERMITTENT or CONTINUOUS. If the CF operates intermittently, then there will be times when it does not use the channel, and thus the channel could be shared with another intermittently operating CF. However, if one of the CFs involved with the channel operates continuously, then that CF must have a sharing capacity of "SHARABLE" to share the channel.
   - *Priority* indicates the level of the CF's priority when using a shared channel. If one of the CFs involved with the channel has a priority of URGENT, then no other CF with a similar priority can share the channel.

Appendix F

Metric Scales

This Appendix discusses some of the types of scales that are available and their applicability to the CAP.

F.1 Scales

As noted in Chapter 6, a metric is a defined measurement method and scale (ISO & IEC, 1998a). One measurement method of a CAP is to compare it with other CAPs.

The scale type of a metric determines what kind of meaningful statements can be made about it and, in particular, what operations can be performed with it. Fenton and Pfleeger describe five well-known scale types: nominal, ordinal, interval, ratio, and absolute (1997). Four of these scales are summarized in Table F.1.

Table F.1: A Classification of Scales (adapted from Stevens, 1959)

Scale	**Operation**	**Location**	**Dispersion**
Nominal	equality	mode	
Ordinal	greater or less	median	percentiles
Interval	distance	arithmetic mean	standard deviation
Ratio	ratio	geometric mean	percent variation

What follows is a discussion of these scales and whether they are appropriate to the CAP.

F.1.1 Nominal Scale

The nominal scale is the most basic metric. In the nominal scale, entities are placed into defined categories or classes, which have no notion of ordering or magnitude, based on some attribute value (Fenton & Pfleeger, 1997). This scale uses semantic expressions to represent objects (teams) for the purpose of identification (referential value). Giving unique and unequivocal names to concepts and defining technical terms also belongs to this scale (Pandian, 2004).

An example of a simple nominal scale metric is the "YES" and "NO" associated with a checklist (Fenton & Pfleeger, 1997). This approach is widely used as a means of evaluating the degree of compliance to an ISO 9241 series standard.

This scale is important because it provides a meaningful way to label entities. One way this scale is used in this Thesis is to define technical terms within the CAP model. Thus, an example of the nominal scale is the use of labels such as "AT" and "System" to describe components within a CAP.

A metric based on the nominal scale alone cannot be used to evaluate accessibility. Under this scale, acquiring a measure of accessibility would involve experimentation with different users. However, because of user differences and multiple interactions among the various factors in the context of the experiment, the results would not be generalizable to other users, even if these factors remain constant over time.

F.1.2 Ordinal Scale

The ordinal scale often augments the nominal scale by providing a ranking to the classification scheme and the ability to combine classes of entities (assuming the combination makes sense with respect to the ordering) (Fenton & Pfleeger, 1997).

Using an ordinal scale amounts to assessing the value of a measured entity and then placing all measured entities according to the ranking scheme. Both value and order can be expressed using words or symbols (Pandian, 2004).

An example of the ordinal scale are surveys that measure attitudes and satisfaction using Likert scales (e.g., the Questionnaire for User Interface Satisfaction measures "satisfaction" on a ten-point Likert scale, see Chin, Diehl, & Norman, 1988).

The ordinal scale is important because it provides a meaningful way to order entities such as judgements about things. One way this scale is used in this Thesis is to define descriptors within the UARM model. Thus, an example of the ordinal scale is the use of descriptors of accessibility such as "fully accessible", "partially accessible", "somewhat accessible", and "not accessible".

F.1.3 Interval Scale

The interval scale captures information about the size of the intervals that separate the classes such that one can compute the difference between any two of the ordered classes. Like the ordinal scale, the interval scale preserves order, but it also allows the preservation of difference. Because there is no rational zero in this scale (i.e., you cannot measure a total lack of the attribute), only addition and subtraction are acceptable operations (Fenton & Pfleeger, 1997). Thus, the interval scale is an arbitrary scale, used for perceiving increments, not ratios. (Pandian, 2004).

An example of the interval scale is measuring a room's temperature in Celsius. Since time is on an interval scale, time-based metrics (e.g., time to perform a task in seconds) are another type of frequently used metric on an interval scale. The interval scale is important because it provides meaningful distances between rankings.

Currently, there is no widely accepted metric of accessibility based on the interval scale. Such a metric would allow showing incremental improvements of accessibility as well as clear differences between measured entities. Most currently accepted metrics of accessibility are checklists (e.g., Chisholm et al., 1999; IBM Corporation, 2002). There are also several Likert scale based accessibility metrics in the literature; however, none have gained a satisfactory level of acceptance (Sloan, Gregor, Rowan, & Booth, 2000).

While the interval scale can be used to document the various attributes of CAP components (e.g., a tactile device using heat might be documented in terms of Celsius), it is not appropriate for the CAP as a whole. Application of this scale to the CAP would require it to be a vector metric.

F.1.4 Ratio Scale

The ratio scale is considered the most useful scale of measurement and is commonly used in the physical sciences, but rarely seen in the social sciences (e.g., a person can have zero income, but it is impossible to have zero attitudes on things). The distinctive feature of a ratio scale is that it has an origin defined by a dominating substantive theory (Stevens, 1959). It provides a rational zero element, units of measure, magnitude, and the meaningful application of ratio to the measure. Since units of measure occur where the measurement mapping must start at zero and increase in equal intervals, all arithmetic operations can be meaningfully applied in the ratio scale (Fenton & Pfleeger, 1997).

This scale makes it possible to have objective measures (e.g., a room's temperature can be measured in Kelvin). However, the application of this scale to concepts such as software accessibility is difficult.

The ratio scale may not be appropriate for metrics of accessibility. The accessibility of a system can be highly subjective and is largely based on human difference. While, it is possible to get absolute zero (e.g., zero accessibility) as well as an upper extreme (e.g., full accessibility) which could be measured as "one", an accessibility measure of 0.54 (i.e., some partial degree of accessibility) has no meaning. There is a problem in that there is currently no way to determine the correct intervals to use when measuring degree of accessibility. Even if the problem of how to measure accessibility on a ratio scale could be solved, the ratio scale may still not be appropriate for any accessibility metric because ISO 9241-11 essentially forbids the use of a metric of "usability units" to measure software usability (ISO, 1998a). Finally, given how difficult it is within the social sciences to develop ratio scale measures without the use of complex statistics, it may not be possible to create an accessibility metric within the ratio scale. For these reasons, although the ratio scale can be used to document the various attributes of CAP components (e.g., a tactile device using heat might be documented in terms of Kelvin), it is not appropriate for the CAP as a whole.

F.1.5 Absolute Scale

The term "absolute scale" was first coined by Stevens (1959) to describe a variation on the ratio scale[1]. This scale is essentially a *count* of the number of elements in the entity set (Fenton & Pfleeger, 1997). It is a unique and unambiguous scale (Pandian, 2004).

The absolute scale is important because counts have multiple uses. A count may indicate a potential problem where further research may be needed to determine its significance. A count may indicate the presence or absence of an entity. Counts have the advantage of being discrete (i.e., only integers can result), having a real zero, having meaningful ratios (e.g., $2x$ does mean twice the quantity) and using nonarbitrary units (i.e., "one" cannot be reset).

Because one can count just about anything, there are dozens of likely examples of the absolute scale that can be derived from the CAP. For example:

- A count of the number of media types described in one CAP may prove useful as a metric to determine the degree a particular system can present content in different ways.
- When comparing CAPs, a count of the number of incompatibilities between them may prove useful as a metric of compatibility (and consequently accessibility).
- A count of the number of ICs in a given CAP may be "good" in helping to understand all of the ways a user can use a given system or it may be "bad" by suggesting that a user may have to contend with more different kinds of media than they might be able to.

Thus, only metrics on the absolute scale can meaningfully generate numbers. However not all of the possible counts are meaningful. For example, a count of the number of lines in a CAP may be useful as a metric of size, but may not prove useful as a metric of access.

In future, when comparing CAPs, it will be important to realize that, according to the CAP model, all incompatibilities are equally significant, but have different costs either in terms of usability or in terms of money. For any count to be meaningful, it must count equally-valued objects. Thus, objects with multiple dimensions, such as CAPs, become that much more difficult to count. For example, a system whose CAP shows twenty incompatibilities may be more accessible to a specific user than one whose CAP shows only one incompatibility. This is the case, if and only if, the system with only one incompatibility exhibits a grossly inaccessible characteristic, such as being unable to output text in a manner accessible to the user. By comparison, the system with twenty incompatibilities, may still allow the user to interact with it to some, albeit limited, degree. Given the importance of text output over all other media, such an incompatibility has a severe accessibility cost.

The monetary costs of a CAP's incompatibility is based on the cost of the AT (or other system components) needed to meet the user's needs. Dealing with cost and assessing cost is future work, because it requires completion of three steps:

1. research on comparing CAPs,
2. development of a database of ATs (and other system components) described in terms of CAPs, and
3. adding AT cost information to the above mentioned database.

Thus, the absolute scale has great potential when applied to whole CAPs.

[1]The primary difference between the two is that the absolute scale does not allow for transformation.

F.1.6 Nominal-Categorical Scale

Pandian (2004) further adds the Typological (or Nominal-categorical) scale to Steven's (1959) list. The Typological scale of measurement identifies types or categories in entities that have been already recognized and named. In the Typological scale, measurement is equivalent to categorization. In Table F.1, this scale would fall between the nominal and ordinal scales because it does not name or rank an attribute, it only categorizes.

A trivial example would be separating a herd of cows from a herd of elephants based on their "cow-ness" or "elephant-ness" (Pandian, 2004).

This scale is important because it provides a meaningful way to categorize or group labelled entities. One way this scale is used in this Thesis is to define groupings of similar characteristics within the CAP model. For example, written text, pictures, and music can all be grouped under "*MediaType*". Written text, spoken text, Braille, and signed text can all be grouped as types of "text". CAPs contain a group of ICs, which can be categorized based on IC type (e.g., user, AT, system, environment). The simplest CAP is just a list of named ICs. Thus, the CAP sorts named entities into categories of information and provides a container in which these categories can be grouped as ICs.

In summary, it appears that several scales are meaningful for the measurement of attributes documented within a CAP, but only one scale — the absolute scale — is meaningful when applied to whole CAPs and then only in limited cases, such as counting incompatibilities derived from a comparison of CAPs.