Information Modelling and Knowledge Bases XXXVII
M. Menšík et al. (Eds.)

doi:10.3233/FAIA251732

Algorithmic Representation of Accessible Human–IT Artifact Interaction

Juho-Pekka MÄKIPÄÄ[a,b1] and Teemu MÄENPÄÄ[a]
[a] *School of Technology and Innovations, Computing Sciences Department, University of Vaasa, Finland*
[b] *Asia AI Institute (AAII), Faculty of Data Science, Musashino University, Japan*

ORCiD ID: Juho-Pekka Mäkipää https://orcid.org/0000-0002-2757-8609
Teemu Mäenpää https://orcid.org/0009-0009-8455-2761

Abstract. It is challenging for a computer to comprehend dynamic characteristics and variables of a human user; however, conceptual modeling can serve as a language to describe these dynamic aspects. In this research, we aim to model the human user's varying input and output modalities in human-computer interaction, making the human appear as part of the system architecture. This allows technologies to adapt to the user's different input and output modalities. As a first step, we model a class diagram to depict a systemic view of interaction. Secondly, we formulated a state machine diagram to describe the behavior of the user and IT artifact in the context of accessible interaction. These models contribute to accessible interaction between humans and computers, where the computer either changes or adjusts its own input and output modalities so that the human can receive them and provide their own outputs according to their abilities.

Keywords. Accessibility; Unified Modeling Language, Human-Computer Interaction

1. Introduction

When designing an accessible user interface, we can determine the state of accessibility by testing it with a user. The basic principle is that if the user can use the interface and perform its functions, it can be considered accessible for that user. At this moment, users have their current abilities (bio-psycho-social) with certain levels. The complex collection of human abilities is sensitive and changeable. Additionally, changes in the interface's features, its executable functions, and the external context affect the user's capabilities to perform tasks. Therefore, we cannot assume that one test result of the interface's accessibility at one moment will still be valid over time, as changes affecting human abilities can occur. Predicting the success of use is based on and focuses on current states and the identification of those states, which accumulate over time. It can be argued, however, that predictions could be more accurate if they were based on examining and measuring changes between current states.

[1] Corresponding Author: Juho-Pekka Mäkipää, juho-pekka.makipaa@uwasa.fi

In this study accessibility is defined by the International Organization for Standardization (ISO), which produces standards for the development of products and processes. ISO defines product accessibility as follows:

> "extent to which products, systems, services, environments and facilities can be used by people from a population with the widest range of user needs, characteristics and capabilities to achieve identified goals in identified contexts of use"[1].

In this study, we aim to model the human user's varying input and output modalities in human-computer interaction (HCI), making the human appear as part of the system architecture. Having a conceptual understanding on how IT artifacts should interact with user's variable abilities gives us a stronger knowledge base to create the IT artifacts that can be adaptable with human factors i.e., an accessible user interface.

Unified Modelling Language (UML) offers diagrams for describing the structure and the behavior of a system [2]. It provides a language for system requirements, design, and implementation that individuals involved in the development process can use for effective communication [2]. It is a tool to describe aspects, objects, and actions involved in system's use [3]. In addition, conceptual modelling can also be a language to advance machine learning [4,5].

In this study, as the first step, we gathered the steps of the HCI process from previous theories, as well as information related to human abilities and characteristics. Next, we used a class diagram to describe the structure of the human user's abilities in HCI context. Then, we formulated a state machine diagram to describe behavior of human and IT artifact in accessible interaction. These models contribute to accessible interaction between humans and computers, where the computer either changes or adjusts its own input and output modalities so that the human can receive them and provide their own outputs according to their abilities. This allows technologies to adapt to the user's different input and output modalities.

This paper is structured as follows: Chapter 2 presents the related literature on accessibility. Chapter 3 presents background for UML modeling and HCI. Our preliminary models are presented in Chapter 4. In Chapter 5, we discuss the research contributions and future research directions.

2. Related Work

HCI designers collect user needs (scenarios, task models, etc.) and software engineers convert them to (class diagram, sequence diagram, code etc.) [6]. Common design models, like the system modeling language, often depict human users and operators as external actors. However, human users should be considered as integral parts of the system [7]. Interaction and multimodality between humans and computers, presented through the UML language and incorporated into the design process, can make requirements easier to understand for a wide range of software engineers [8,9]. For example, Ordoñez-Briceño et al. [10] introduced a UML-based streamline for developers to incorporate Web Content Accessibility Guidelines (WCAG) guidelines. General standards and guidelines, such as the WCAG, guide the development of accessible websites. However, applying these guidelines in development requires specific knowledge of WCAG practices.

UML can also work as a tool for risk analysis to understand and handle concepts of human factors in a use of computer [11]. Considering humans in the context of IT artifact use, there are several affecting factors [12]. First, human abilities vary in terms of sensory perceptions, cognition, and functional operations [13–18]. Secondly, users perceive the task differently and characteristics of a specific task can significantly influence how users perceive the task. Thirdly, the context of use varies in terms of environmental factors, users' emotional state, socio-cultural factors, and socio-technical factors wherein the cultural, political, sociological, and historical aspects of that context influence the user [19]. Taking user's emotional state as a one example of contextual factor, humans can experience an extensive range of emotions—over 34,000 unique ones, according to Plutchik (2001) [20] . To make them more concise, these emotions can be distilled into eight primary categories: anger, fear, sadness, joy, disgust, surprise, trust, and anticipation.

In addition to human, task, and contextual characteristics, according to Hofstede et al. (2008) [21], "human is social." Human beings live within various social groups. Each person belongs to multiple groups, such as families, villages, regions, countries, religious communities, organizations, and companies [21]. The existence of a group implies the presence of an outer boundary. Some cultures emphasize strict and well-defined group identities, while others have more fluid boundaries. However, the reasons for these variations remain speculative. From an evolutionary perspective, humans occupy a middle ground between solitary existence and perfect group cohesion. Consequently, the question of whether we belong to the same group is relevant in nearly all social interactions.

The fact is we humans are different both externally and internally. In 2001, the World Health Assembly [13], classified a person's functional abilities to understand and follow the large and complex scale of human psychological and physical characteristics. The International Classification of Functioning, Disability, and Health (ICF) serves as a tool to define and measure human functioning in society, with a focus on health, functioning, and a person's abilities. The ICF presents an integrated model of disability that considers both medical and social perspectives, including biological, psychological, and social aspects (biopsychosocial model). The medical model defines disability as a feature of the person caused by disease, trauma, or other health conditions requiring medical treatment to 'heal' the individual [14]. In contrast, the social model views disability as a socially created problem resulting from an unaccommodating environment that neglects the rights of persons with disabilities [14]. Similarly, the Disability Rights Movement recognizes that an unaccommodating environment or technology arises from neglecting the rights of people with disabilities, thus framing disability as socially constructed. According to Amundson and Tresky (2007) [22], disability represents an interaction between biomedical conditions (impairments) and the social environment, rather than a straightforward property of an individual.

Human abilities vary, primarily based on the severity of their impairments. Additionally, the same individual often experiences varying degrees of constraint related to sensory perception, cognition, and mobility. For instance, approximately 40% of accessibility-related studies focus on people with visual impairments but rarely consider other characteristics of these participants, as the primary focus remains on visual disabilities during that specific time [23]. Yet, it is natural that the same individuals who participated in those studies may exhibit differences in cognitive abilities, for example. Similarly, there are studies that delve into cognitive matters, mobility, limb movement, or complex disorders, but rarely focus on phenomena around their changes.

In the context of HCI, human actions taken with computer, can shift from stability to instability, and can be explained and described in non-linear processes [24]. For example, the process philosophy sees nature in an everlasting dynamic process where change is the only constant truth that can be identified, measured, and justified [25]. When it comes to technology adoption, it is neither static nor linear [26]. Instead, the use of technology unfolds as a temporal, changing phenomenon. It occurs in sequences or iterative loops of interest.

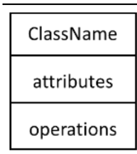
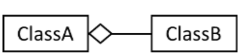
In summary, human factors are intricate, influenced by context, emotions, and dynamic processes. Understanding this complexity is crucial for accessible design and system development. If we think that human users are not in the status quo, then the case is we can only observe human behavior or change. Humanity also includes the fact that individuals can change their own behavior, either intentionally, unintentionally, or due to an external influence, in which case each use of an IT artifact in relation to time is different, i.e., dynamic.

3. Core Principles of UML and HCI

UML offers diagrams to describe, for example the structure system or the behavior of objects within the system [2]. The class diagram is one of the most widely used UML diagrams, applied in various phases in the software development process [27]. The level of abstraction in the class diagram can vary depending on the phase. In this study, we use the class diagram to model the static structure of a human user and its variables in their perceptual, cognitive, functional abilities. Thus, we describe the objects of a human user and the relationships between them. However, these objects and the relationships between them are described in static mode which do not change over time. With the class diagram, we aim to create a conceptual view of the user's abilities and their relationships.

During their lifecycle, objects go through different phases. Behavioral diagrams can be used to define an object's behavior in detail [2]. Behavior can be defined through the action of a single object or caused by interactions between multiple objects. The behavior of objects can be represented in UML using a state machine diagram [2]. This diagram shows an object's behavior in different states and state transitions. With the state machine diagram, we aim to create a conceptual view of the HCI, focusing on the input and output modalities of both the human and the computer. Their interaction depends on the human's ability to receive the computer's output and provide their own input. In Table 1 and Table 2, we describe notations applied in this study.

Table 1. Notations and Descriptions Applied in Class Diagram [2].

Notation	Description
	Class provides a description of features and behavior of a set of objects
	Association describes a connection or relationship between classes.
	Shared aggregation is a special association that describes parts-whole relationship between classes. ClassB is part of ClassA.

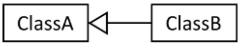
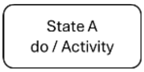
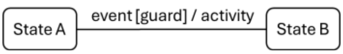
	Generalization depicts inheritance relationship. ClassB inherits attributes and operations from ClassA. Furthermore, ClassB can have its own attributes and operations.

Table 2. Notations and Descriptions Applied in State Machine Diagram [2].

Notation	**Description**
	State describes a specific time span during the object's life cycle. When an object is in a certain state, activities are executed. Do-activity refers to an activity that is executed while the object is in this state.
	Transition depicts the situation when object's state change from source state (State A) to target state (State B). Typically, events trigger the state transition. If the transition does not have event and guard, the transition is automatic which can take place as soon as do-activities of source state are completed. Guards can be used to control the transitions. Guard is a Boolean expression that is evaluated. If the guard is true, transition occurs. During the transition activities can be executed. For example, an event can be sent (denoted with keyword send) to another object. The sent event then activates the transitions in the receiving object's state machine diagram.
	Initial state depicts the start of the state machine diagram.

3.1. Variables and Relationships Between Components of Accessibility

To model human factors in this study, we retrieved information related to HCI process and user's abilities from prior research e.g. [12,15,16,18,19,28]. Figure 1 illustrates the basic elements of HCI including human user and IT artifact.

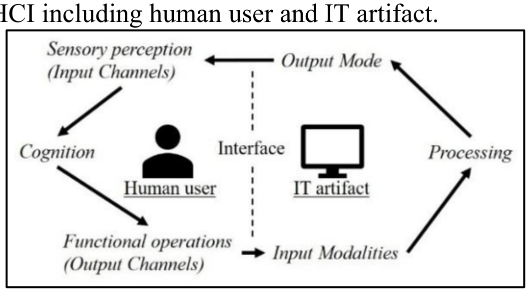


Figure 1. Basic HCI Model [18,28].

In HCI, the user detects the features of IT artifact (*content, presentational style, functionality, interactional style, and structure*) with sensory perception [29,30] having varying abilities in sight, hearing, touch, smell, taste, and balance [14,18,31].

The features of IT artifact can be revealed by some of the IT artifact's output modalities (*visual, auditive, tactile, olfactory, gustatory, or vestibular*) [18]. In practice, this means texts, images, videos, graphs, charts, tables, shapes, etc. and these have differences in their presentational style, for example, use of colors, size, shapes, etc.

After perceiving the IT artifact output, human cognition interprets perceived input and guides human body functions i.e., functional operations [18,28,32]. Domains of users' cognition can be classified and unified basing on the CHC theory by McGrew, (2009) [17], and the ICF classification of human ability to apply knowledge [31] as follows: Focusing attention, Memory, Thinking and speed of processing, Reading and writing, Mental functions of language, Calculating and quantitative knowledge, Solving problems, Making decisions and reaction speed, Psychomotor functions and sequencing complex movements and speed, Emotional functions, Perceptual functions, Higher-level cognitive functions and domain-specific knowledge, Experience of self and time functions, Comprehension-knowledge.

Functional operations as a human output (*movements, voice, and sight*) [33,34] work with IT artifact's input modalities which require either *movements, force, sound, or images* as an input mode [18,28,32].

4. Towards an Algorithmic Model of Human-IT Artifact Interaction

As a first step, we model a class diagram to depict the structure of the human user's abilities. Secondly, we devised a state machine diagram to describe behavior of user and IT artifact in the context of accessible interaction.

4.1. Class Diagram of Systemic View of Interaction

In Figure 2, we describe a static model of human user and IT artifact as a system that is structured with classes (objects), their attributes, operations, and the relationships between objects.

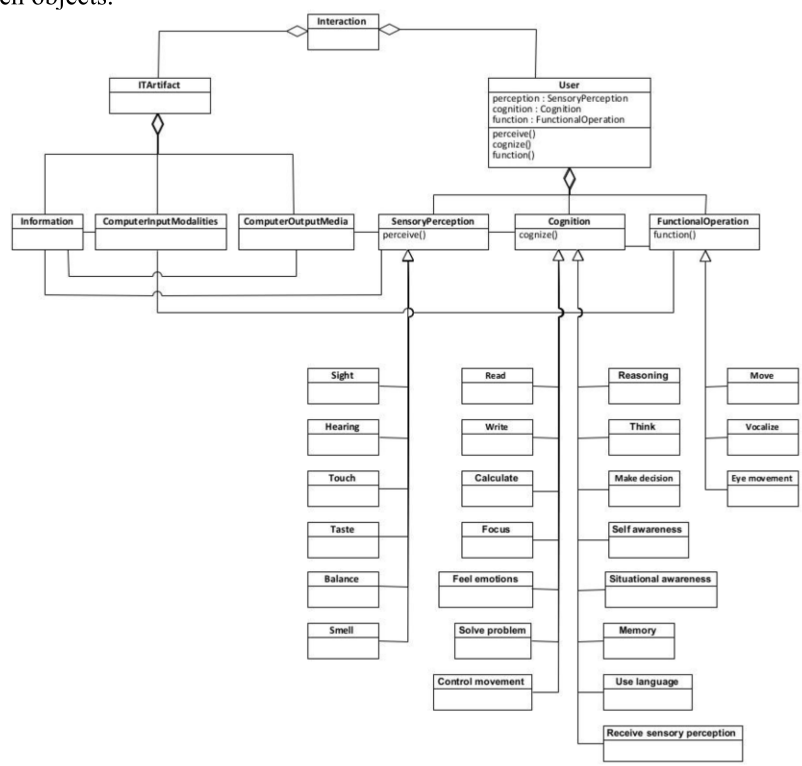


Figure 2. Class Diagram of Systemic View of Interaction

The relationships between classes and their operations are described below in detail.

In the domain of human user, *User* class delegates tasks to aggregated classes: *SensoryPerception*; *Cognition*; and *FunctionalOperation*. SensoryPerception class then forwards or delegates the execution of the given task, i.e., perception, to a corresponding subclass *Sight*; *Hearing*; *Touch*; *Smell*; *Taste*; or *Balance.* For example, an auditory impulse is delegated to be received by Hearing.

Class *Cognition* delegates the task to a corresponding subclass responsible for executing the task. For example, an impulse in text format is delegated to be executed by class *Read.* Subclasses of class *Cognition* represent different cognitive functions, and they are given representative class names in the class diagram. The names of the subclasses are:

- Reading and writing=Read; Write;
- Calculating and quantitative knowledge=Calculate;
- Focusing attention=Focus;
- Emotional functions=Feel emotion;
- Solving problems=Solve problem;
- Psychomotor functions and sequencing complex movements and speed=Control movement;
- Higher-level cognitive functions and domain-specific knowledge=Reasoning;
- Thinking and speed of processing=Think;
- Making decisions and reaction speed=Make decision;
- Experience of self and time functions=Self-awareness;
- Comprehension-knowledge=Situational awareness;
- Memory=Memory;
- Mental functions of language=Use language; and
- Perceptual functions=Receive sensory perception.

FunctionalOperation class delegates the task to a corresponding subclass. The subclasses of FunctionalOperation are named with representative class name in the class diagram as follows: *Movements*=Move*; Voice*=Vocalize*; and Sight*=Eye movement. FunctionalOperation class directs class ComputerOutputMedia. Interaction, for example, requires the pressing of a button, at which point class FunctionalOperation gives the task execution to class Move.

Class ITArtifact comprises of three classes: Information; ComputerOutputMedia; and ComputerInputModalities. Information class depicts the messages and signals to be conveyed to user. SensoryPerception class is associated with class Information; and class ComputerOutputMedia. Association between the classes can be interpreted as follows human sensory perception perceives the information via some of the computer output media. Information class is associated with class ComputerOutputMedia. Association depicts a situation where some computer output media presents the contents of Information class. Output media can be *visual, auditive, tactile, olfactory, gustatory, or vestibular.* ComputerInputModalities class is associated with class FunctionalOperations. Association represents the situation when the user is required to act by using functional operations to make actions with some of the computer input modalities: *movements, force, sound, showing images.*

4.2. Modeling Accessible Behavior of User and IT Artifact

The concept of accessibility is universal in nature. It desires the extent where any user regardless of its characteristics in abilities can achieve the extent of where this user is capable to operate with IT artifact [19].

In Figures 3 and 4, we illustrate how the IT artifact should behave while serving the user depending on its various characteristics. The state diagram depicts the user's behavior, as a part of the operation of the IT artifact. The case in the figure illustrates a situation where the user employs the IT artifact to perform a specific task. During the interaction, the user perceives and interprets the inputs provided by the IT artifact and responds accordingly, thereby generating their own output for the IT artifact. The IT artifact, in turn, provides output and interprets whether the user receives it or not. If the user does not perceive the given output, the IT artifact adjusts or changes its mode until the user has received the output. Similarly, the IT artifact waits for input from the user until the user provides it. If the user cannot provide the required input, the IT artifact adjusts or changes the input mode until the user can provide it.

Figure 3 describes user's behavior while interacting with IT artifacts.

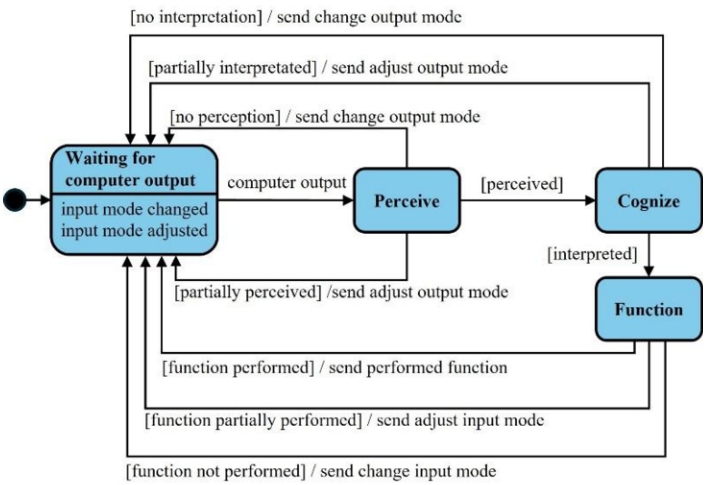


Figure 3. A State Machine Diagram of Accessible Interaction Between User and IT Artifact (User Behavior).

User's behavior is represented with four states; Waiting for computer output, perceive, cognize, and function. State transitions are triggered with the following events:

Event *computer output* causes transition from state *Waiting for computer output* to state *Perceive* with guard condition *perceived.* In practice this means user perceive or not perceive IT artifact's output. In a case where user perceives IT artifact's output successfully, *Perceived* guard condition controls state transition from *Perceive* to *Cognize*. If user has perceived computer output partially, action "send adjust output mode" is executed. If user has not perceived computer output, action "send change output mode" is executed. In practice, computer output is adjusted or changed while user perceives it.

In state *Cognize,* guard condition *interpreted* controls state transition to state *Function.* If user has interpreted perceived computer output partially, action "send adjust output mode" is executed. If user has not interpreted computer output, action "send change output mode" is executed. In practice, computer output is adjusted or changed while user interprets it successfully.

In state *Function,* if the user has performed function successfully, one interaction is successful, and the next interaction loop can start. If user has performed function partially, action "send adjust input mode" is executed. If user has not performed function, action "send change input mode" is executed. In practice, computer input mode is adjusted or changed while user perform function successfully.

The state machine diagram in Figure 3 shows the behavior of the IT artifact in interaction with user. It describes the behavior of an IT artifact in situations where the user either perceives, does not perceive, or partially perceives the output given by the IT artifact, and situations where the user either can, cannot, or can partially provide input to the IT artifact.

Figure 4 describes sequence of operations begins from the state "Waiting for input", where the IT artifact is waiting for input from the user.

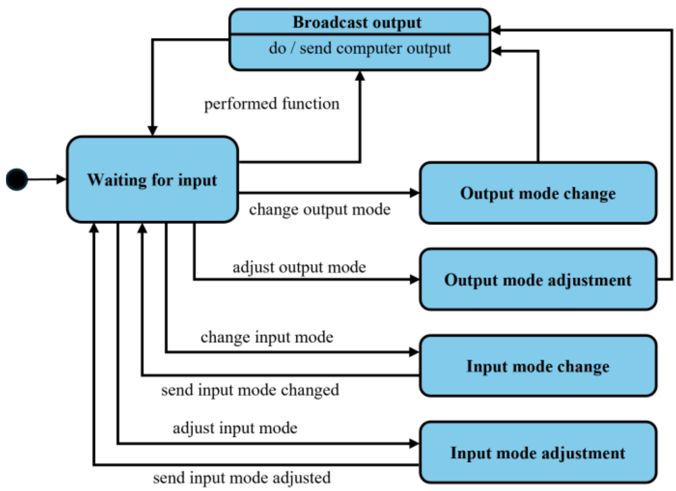


Figure 4. A State Machine Diagram of Accessible Interaction Between User and IT Artifact (IT Artifact Behavior).

IT artifact receives events *performed function*, *change output mode*, *adjust output mode*, *change input mode*, and *adjust input mode* from user's state machine. The send actions occur during state transition that are shown in the user's state diagram. These actions are events that cause changes in IT artifact's states. The state transition triggered by the events are *Waiting for input* to states either *Output mode change, Output mode adjustment, Input mode change,* or *Input mode adjustment* depending on actions sent by user behavior. The states *change output mode*, *adjust output mode* automatically changes to state *Broadcast output* after the output mode has been switched from the previous mode to the new one.

Events *input mode adjusted* and *input mode changed* are sent to user. Events are modeled as internal transitions in state *Waiting for computer output.* Through internal transitions, we demonstrate that the user has noted changes in the input mode.

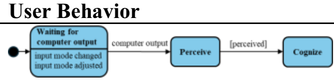
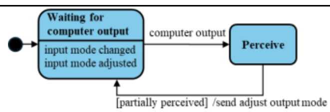
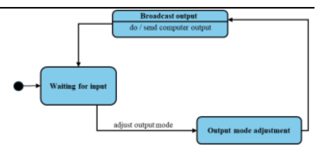
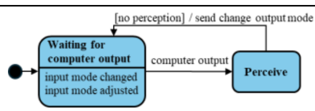
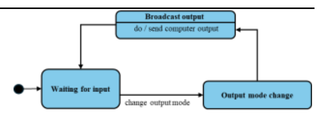
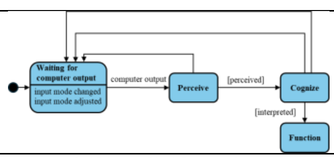
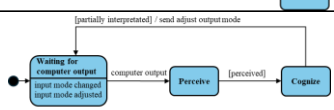
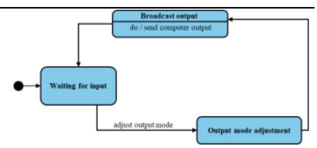
As a practical example, when the user does not detect the output sent by the computer, the system receives a command to change the output mode. When the system receives the command to change the output mode, the output mode is switched to the next one. This function repeats until the user perceives the output. If the user perceives the output only partially, a command is sent to adjust the output accordingly. The output is adjusted until the user perceives it. For example, the user perceives the computer's output in text form but cannot read it due to the small font size. In this case, the font size is adjusted until the user can read the text.

5. Scenario-Based Analysis of Proposed State Diagrams

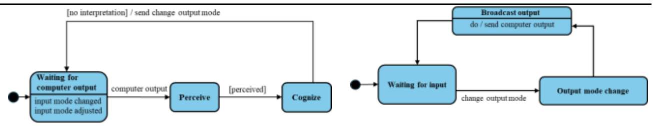
In Table 3, we present a scenario-based examination of the state diagram through nine example cases. This approach applies scenario-based logical reasoning [35] to evaluate the feasibility of our conceptual models. We described scenarios to simulate the behavior of user and IT artifact interaction in scenarios where (1) user perceives, (2) partially perceives, or (3) do not perceive IT artifact's output. Next, in scenarios where user (4) can interpret perceived output, (5) can interpret partially, and (6) cannot interpret. Then, in cases where user (7) can perform functions, (8) can perform functions partially, and (9) cannot perform functions.

As a fictive example case, the application prompts the user with a pop-up window asking whether they want to save a file. The user interacts with the system using a computer and a mouse.

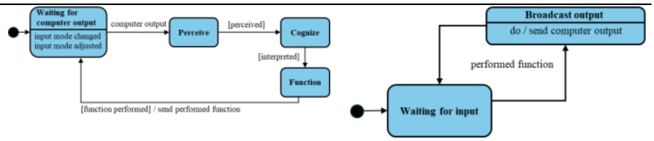
Table 3. Example Scenarios

Scenario	User Behavior	IT Artifact Behavior
Scenario 1 (Perceived): The user is able to read the text, after which they begin to consider what should be done.		No action
Scenario 2 (Partially perceived): The user notices the text but can only partially recognize it. The application increases the font size of the text.		
Scenario 3 (No perception): The user is unable to read the text in the pop-up window because they cannot see it. The application converts the text into speech, allowing the user to receive the information audibly.		
Scenario 4 (Interpreted): The user understands the text and then begins clicking the 'Yes' button.		No action
Scenario 5 (Partially interpreted): The user does not understand the text in the pop-up window because they are unfamiliar with certain terminology. The application reformulates the content into a more easily understandable format.		

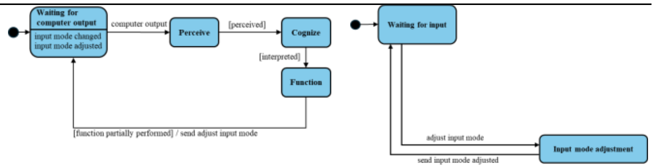
Scenario 6 (No interpretation): The user does not understand the text at all. The application transforms the message into a visual format.



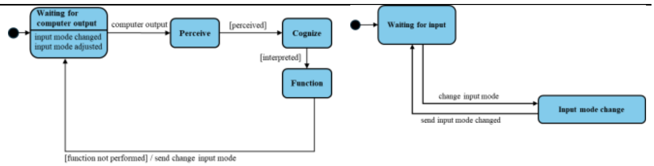
Scenario 7 (Function performed): The user performs the action by clicking the 'Yes' button. The application proceeds to the next step.



Scenario 8 (Function partially performed): The user attempts to click the 'Yes' button with the mouse but struggles to hit it accurately due to fast cursor movement and the small size of the button. The application responds by reducing mouse sensitivity and enlarging the button size.



Scenario 9 (Function not performed): The user is unable to use mouse or keyboard buttons. The application switches to voice control.



6. Discussion and Concluding Remarks

Considering users as natural beings, with a changing and limited range of abilities (bio-psycho-social), they set requirements for technologies to be adaptable to the user's different input and output modalities. To design such a system, detaching human abilities conceptually enables the design of IT artifacts that are fundamentally stable but can vary in their existence. An IT artifact, being artificial, can be created with stability at its core. Physical and informational resources basically operate predictably. However, human users can change their behavior.

In this study, we modelled human user not as an external actor but as integral part of system [7]. UML as tool to describe humans as an integral part of system enables software engineers to adapt behavior of user into system design easier [6,8,9]. The UML model of a human user serves as a tool for developers to identify varying abilities of humans to use the system and to guide the design work in making decisions that benefit users with disabilities. According to Guiochet et al. [11], UML-based models can also work as a tool for risk analysis.

This study contributes to theories of HCI by presenting a structured description of HCI. As a first step, we used a class diagram to describe the structure of human users in HCI. Secondly, with the help of a state machine diagram, we described the behaviors of human users and IT artifacts in an interaction. Specifically, we detailed how IT artifacts adapt to users' varying abilities to perceive, cognize, and act with the IT artifact’s input and output modalities. These models contribute to accessible interaction between humans and computers, where the computer changes or adjusts its own input and output modalities so that the human can receive them and provide their own outputs according to their abilities.

As a second contribution, this research contributes to the studies that examine the adaptability and responsiveness of an IT artifact (change) instead of its current state. Often, accessibility is studied by testing the current state of an IT artifact and how the user interacts with it. Examining the current state of the IT artifact and the factors affecting its use is important. However, if we want to develop the adaptability and responsiveness of systems to the varying characteristics of users, understanding of the change between states and these phenomena can be incorporated into the design work is essential.

Attempts to predict how systems are accessible to all users often rely on verifying the current state of interaction and resolving its issues. However, user characteristics change, and the previously identified current state no longer applies over time. Predictions could become more accurate if they were based on changes between current states. In addition to combining the information obtained from current states, if we also combine the information obtained from changes, we have a stronger knowledge base to create the next version of an accessible user interface, which is likely to also consider probable future changes.

To simplify prediction where the change is a driver, IT artifact's current extent (Initial Value) changes to new extent (Final Value), which becomes Current extent (Initial Value2) for next change, and for next new extent (Final Value2) and so on. Therefore, the change can be measured as its simplest form as follows:

$$Change = Final\ Value - Initial\ Value$$

The change could be studied from the process philosophy viewpoint. According to the process philosophy, there is no stable state, but a continuous process prevails everywhere, i.e. a state of change. Similarly, in the use of an IT artifact, the user is affected by processes including also changes in him/herself, the environment, and in the IT artifact itself affecting the usage situation. This means one can develop an IT artifact that is easier to use in one specific context testing the use of the IT artifact in that context. However, information gained from the tests of an IT artifact may be invalid, or it may be outdated when changes occur in the context, for example, in user or in the environment. There are still issues that need studying, for instance, human emotions, which are evolutionarily rooted [20], and they exhibit diverse variations across personalities and contexts. These emotions interact with factors like culture and emotions. In this study, we omitted certain factors related to HCI. We excluded the context of use even if it may influence also for users' abilities thought the user's emotional state, socio-cultural factors, socio-technical factors, cultural, political, sociological, and historical factors of that context [14,33,36–38]. Moreover, users' expectations for IT artifact, past experiences, prejudices, evoked memories, unmet expectations, and confidence [39] are also left out from analysis. We also omitted task characteristics, such as how complex, motivating, or engaging the task in HCI is for user [19]. These factors were left out in this study since they are subjective in nature. Therefore, perceptions of the context and the task should be experimentally examined with users thought different scenarios.

We recommend for the next questions for future topics to include how IT artifacts can and should detect that user is not perceiving or is perceiving partially the output. Which are relevant/appropriate indicators and parameters for the events that cause transitions to change/adjust the output? And how the context and characteristics of task influence the interaction between user and IT artifact.

In conclusion, instead of creating general user personas, we should focus on more understanding complexity and changes in the interaction between humans and IT artifacts. Users should be considered as unique entities which are complex, unknown, and unpredictable. Therefore, the future research should develop methods to better understand continuum of process-based complex behavior in HCI that considers humans natural (biological), psychological, complex social, and technical complexities (culture, learning etc.) as an unstable factor of HCI.

References

[1] International Organization for Standardization, ISO 9241-11:2018, Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts, (2018). https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en (accessed September 12, 2020).

[2] M. Seidl, M. Scholz, C. Huemer, and G. Kappel, UML @ Classroom: An Introduction to Object-Oriented Modeling, Springer International Publishing, Cham, 2015. doi:10.1007/978-3-319-12742-2.

[3] C. Moral, A. De Antonio, and X. Ferre, A visual UML-based conceptual model of information-seeking by computer science researchers, *Information Processing & Management*. **53** (2017) 963–988. doi:10.1016/j.ipm.2016.10.005.

[4] R. Lukyanenko, A. Castellanos, J. Parsons, M. Chiarini Tremblay, and V.C. Storey, Using Conceptual Modeling to Support Machine Learning, in: C. Cappiello, and M. Ruiz (Eds.), Information Systems Engineering in Responsible Information Systems, Springer International Publishing, Cham, 2019: pp. 170–181. doi:10.1007/978-3-030-21297-1_15.

[5] R. Lukyanenko, A. Castellanos, V.C. Storey, A. Castillo, M.C. Tremblay, and J. Parsons, Superimposition: Augmenting Machine Learning Outputs with Conceptual Models for Explainable AI, in: G. Grossmann, and S. Ram (Eds.), Advances in Conceptual Modeling, Springer International Publishing, Cham, 2020: pp. 26–34. doi:10.1007/978-3-030-65847-2_3.

[6] M.G. De Paula, S.D.J. Barbosa, and C.J.P. De Lucena, Conveying human-computer interaction concerns to software engineers through an interaction model, in: Proceedings of the 2005 Latin American Conference on Human-Computer Interaction - CLIHC '05, ACM Press, Cuernavaca, Mexico, 2005: pp. 109–119. doi:10.1145/1111360.1111371.

[7] M.E. Watson, C.F. Rusnock, J.M. Colombi, and M.E. Miller, Human-Centered Design Using System Modeling Language, *Journal of Cognitive Engineering and Decision Making*. **11** (2017) 252–269. doi:10.1177/1555343417705255.

[8] M. Thomas, I. Mihaela, R.M. Andrianjaka, D.W. Germain, and I. Sorin, Metamodel based approach to generate user interface mockup from UML class diagram, *Procedia Computer Science*. **184** (2021) 779–784. doi:10.1016/j.procs.2021.03.096.

[9] Z. Obrenovic, and D. Starcevic, Modeling multimodal human-computer interaction, *Computer*. **37** (2004) 65–72. doi:10.1109/MC.2004.139.

[10] K. Ordoñez-Briceño, J.R. Hilera, L. de-Marcos, S. Otón-Tortosa, and S. Cueva-Carrión, UML Profile to Model Accessible Web Pages, *IEEE Access*. **12** (2024) 77181–77213. doi:10.1109/ACCESS.2024.3406688.

[11] J. Guiochet, G. Motet, C. Baron, and G. Boy, Toward a Human-Centered UML for Risk Analysis, in: C.W. Johnson, and P. Palanque (Eds.), Human Error, Safety and Systems Development, Springer US, Boston, MA, 2004: pp. 177–191. doi:10.1007/1-4020-8153-7_12.

[12] J.-P. Mäkipää, J. Norrgård, and T. Vartiainen, Factors Affecting the Accessibility of IT Artifacts: A Systematic Review, *CAIS*. **51** (2022) 666–702. doi:10.17705/1CAIS.05129.

[13] World Health Organization, Towards a Common Language for Functioning, Disability and Health ICF. (WHO/EIP/GPE/CAS/01.3), (2002). https://www.who.int/classifications/icf/icfbeginnersguide.pdf?ua=1.

[14] World Health Organization, How to use the ICF: A practical manual for using the International Classification of Functioning, Disability and Health (ICF), (2013).

[15] J.B. Carroll, Human Cognitive Abilities: A Survey of Factor-Analytic Studies, Cambridge University Press, Cambridge, 1993. doi:10.1017/CBO9780511571312.

[16] J. Carroll, Human-computer interaction: Psychology as a science of design, *Annual Review of Psychology*. **48** (1997) 61–83. doi:10.1146/annurev.psych.48.1.61.

[17] K.S. McGrew, CHC theory and the human cognitive abilities project: Standing on the shoulders of the giants of psychometric intelligence research, *Intelligence*. **37** (2009) 1–10. doi:10.1016/j.intell.2008.08.004.

[18] L. Schomaker, S. Münck, and K. Hartung, eds., A Taxonomy of Multimodal Interaction in the Human Information Processing System, *Rep. Esprit Proj.* **8579** (1995).

[19] J.-P. Mäkipää, Explaining Accessibility: Possible Variables in Users' Abilities, Tasks, and Contexts in IT Artefact Use, *THCI*. **15** (2023) 414–441. doi:10.17705/1thci.00096.

[20] R. Plutchik, The Nature of Emotions: Human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice, *American Scientist*. **89** (2001) 344–350.

[21] G.J. Hofstede, C.M. Jonker, and T. Verwaart, Modeling culture in trade: uncertainty avoidance, in: Proceedings of the 2008 Spring Simulation Multiconference, Society for Computer Simulation International, San Diego, CA, USA, 2008: pp. 143–150.

[22] R. Amundson, and S. Tresky, On a Bioethical Challenge to Disability Rights, *Journal of Medicine and Philosophy*. **32** (2007) 541–561. doi:10.1080/03605310701680924.

[23] D.M.B. Paiva, A.P. Freire, and R.P. de Mattos Fortes, Accessibility and Software Engineering Processes: A Systematic Literature Review, *Journal of Systems and Software*. **171** (2021) 110819. doi:10.1016/j.jss.2020.110819.

[24] D. Straussfogel, and C. von Schilling, Systems Theory, in: R. Kitchin, and N. Thrift (Eds.), International Encyclopedia of Human Geography, Elsevier, Oxford, 2009: pp. 151–158. doi:10.1016/B978-008044910-4.00754-9.

[25] J. Seibt, Process Philosophy, *The Stanford Encyclopedia of Philosophy (Spring 2024 Edition), Edward N. Zalta & Uri Nodelman (Eds.)*. (2024). https://plato.stanford.edu/archives/spr2024/entries/process-philosophy (accessed April 5, 2024).

[26] E. Brangier, and S. Adelé, Evolutions in the human technology relationship: rejection, acceptance and technosymbiosis, (2014).

[27] M. Ozkaya, and F. Erata, A survey on the practical use of UML for different software architecture viewpoints, *Information and Software Technology*. **121** (2020) 106275. doi:10.1016/j.infsof.2020.106275.

[28] J.H. Gerlach, and F.-Y. Kuo, Understanding Human-Computer Interaction for Information Systems Design, *MIS Quarterly*. **15** (1991) 527–549. doi:10.2307/249456.

[29] M. Hassenzahl, The Thing and I: Understanding the Relationship Between User and Product, in: M.A. Blythe, K. Overbeeke, A.F. Monk, and P.C. Wright (Eds.), Funology, Springer Netherlands, Dordrecht, 2003: pp. 31–42. doi:10.1007/1-4020-2967-5_4.

[30] W3C, Web Content Accessibility Guidelines (WCAG) 2.1, (2018). https://www.w3.org/TR/WCAG21/ (accessed June 14, 2020).

[31] WHO, ICF Browser, (n.d.). https://apps.who.int/classifications/icfbrowser/ (accessed March 24, 2021).

[32] R. Babu, R. Singh, and J. Ganesh, Understanding Blind Users' Web Accessibility and Usability Problems, *AIS Transactions on Human-Computer Interaction*. **2** (2010) 73–94.

[33] J. McKay, P. Marshall, and R. Hirschheim, The Design Construct in Information Systems Design Science, *Journal of Information Technology*. **27** (2012) 125–139. doi:10.1057/jit.2012.5.

[34] J. Sevilla, G. Herrera, B. Martínez, and F. Alcantud, Web accessibility for individuals with cognitive deficits: A comparative study between an existing commercial Web and its cognitively accessible equivalent, *ACM Trans. Comput.-Hum. Interact.* **14** (2007) 12-es. doi:10.1145/1279700.1279702.

[35] C. Sonnenberg, and J. vom Brocke, Evaluations in the Science of the Artificial – Reconsidering the Build-Evaluate Pattern in Design Science Research, in: K. Peffers, M. Rothenberger, and B. Kuechler (Eds.), Design Science Research in Information Systems. Advances in Theory and Practice, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012: pp. 381–397. doi:10.1007/978-3-642-29863-9_28.

[36] K. Lyytinen, and M. Newman, Explaining information systems change: a punctuated socio-technical change model, *European Journal of Information Systems*. **17** (2008) 589–613. doi:10.1057/ejis.2008.50.

[37] G. Meiselwitz, B. Wentz, and J. Lazar, Universal Usability: Past, Present, and Future., *Foundations and Trends in Human-Computer Interaction*. **3** (2010) 213–333.

[38] H. Sharp, N. Lotz, L. Mbayi-Kwelagobe, M. Woodroffe, D. Rajah, and R. Turugare, Socio-cultural factors and capacity building in Interaction Design: Results of a video diary study in Botswana, *International Journal of Human-Computer Studies*. **135** (2020) 102375. doi:10.1016/j.ijhcs.2019.102375.

[39] A. Aizpurua, M. Arrue, and M. Vigo, Prejudices, memories, expectations and confidence influence experienced accessibility on the Web, *Computers in Human Behavior*. **51** (2015) 152–160. doi:10.1016/j.chb.2015.04.035.