Identifiability: A Predictive Quantitative Measure for the Comparison of a Task Designed in Different Interaction Styles

Georgios Christou[1]

Cyprus College
6 Diogenes str. Engomi,
P.O. Box 22006, 1516,
Nicosia,
Cyprus
gchristou@cycollege.ac.cy

Robert J. K. Jacob

Tufts University
161College Ave.
Medford, MA 02155
USA
jacob@cs.tufts.edu

Abstract

This paper presents a predictive quantitative measure for the comparison of the performance of a task, when that task is designed in different interaction styles. The paper presents an extension of Task Knowledge Structures (Johnson and Johnson, 1991a) and based on this extension it presents identifiability, which is the evaluative and predictive measure that is the main focus of the paper. The paper then presents an experiment, which provides evidence towards the validity of the measure and its intended usage.

1 Introduction

The Direct Manipulation or WIMP (Windows, Icons, Menus, Pointer) user interface (Hutchins, Hollan and Norman, 1986), (Shneiderman, 1983) led to a new generation of interaction styles, which made computer use easier and more widespread than before, and which has become the nearly universal state of practice in user interface design. Today, we see research that looks toward a new generation of non-WIMP (Jacob, 1995) or Post-WIMP (Van Dam, 1997) interaction, but it is appearing on a variety of fronts, less unified than the previous generation. Examples come from new research, in areas such as virtual reality, augmented reality, ubiquitous, pervasive and handheld interaction, tangible user interfaces, lightweight, tacit, or passive interaction, perceptual interfaces, affective computing, context-aware interfaces, and speech and multi-modal interfaces. The number of new interaction styles that have appeared in recent years, makes the problem of creating a better user interface for some task ever greater than before, when the designer had only the choice of direct manipulation to design an interface.

This paper has two goals: the first is to present an extension of Task Knowledge Structures (TKS) (Johnson and Johnson, 1991a) and the second goal of this paper is to provide a way to compare and evaluate the seemingly disparate new interaction styles against each other, and to explicate the qualities that seem to make the tasks easier for users. The authors believe that this second goal can only be achieved with the introduction of measures that are "interface style independent", which means that they evaluate the interaction style from the perspective of the user and how the same task is performed in two interaction styles. It is also important to look at how the usage of the tools and principles of each interaction style makes it easier or harder for the user to create a mental model of the workings of the interface presented to him/her.

The paper presents a measure that may be used to provide a quantitative measurement, which is comparative in use, called Identifiability. This measure takes into account knowledge that the user has about the world. By using an extension of Task Knowledge Structures (Johnson and Johnson, 1988) it measures the fitness of certain pieces of an interface that is designed in a specific interaction style for the accomplishment of a specific task. The measurement can be used comparatively to another measurement about performing the same task in an interface that is designed in a different interaction style.

The following sections provide the background upon which the measure is based on, the process which one should use to perform measurements using this measure, and finally an experiment that the authors believe shows that the measure is valid for its intended purpose.

[1] Also at: Tufts University, 161College Ave., Medford, MA 02155, USA

2 Background

Work on identifiability is based heavily on work done on non-WIMP interaction styles, as well as work on affordance and Task Knowledge Structures. This section presents a summary of this work that is relevant to the concept of identifiability.

2.1 Affordance and Constraint

2.1.1 Affordance

The work on Identifiability is an extension to the work on Affordance (Gibson, 1977, 1979), which was subsequently extended and clarified for the field of Human Computer Interaction (HCI) by Norman (Norman, 1988, 1999), Gaver (Gaver, 1991), McGrenere and Ho (McGrenere and Ho, 2000), Hartson (Hartson, 2003), and others.
Gibson coined the term affordance (Gibson, 1977 and Gibson, 1979), to refer to the actionable properties between the world and an actor (whatever that actor may be) (as cited in Norman, 1999). In Gibson's view, an affordance just exists in the world. It is not a property that must be observable in any way. Norman presented a revised definition for affordance (Norman, 1988), and introduced the concepts of physical affordance and perceived affordance. The latter type of affordance defines the clues that a device or user interface give to the user as to the functionality of an object. The object may be the device itself, or a property or feature of the user interface. Physical affordance on the other hand, refers to the definition of affordance as stated by Gibson (Gibson, 1977).

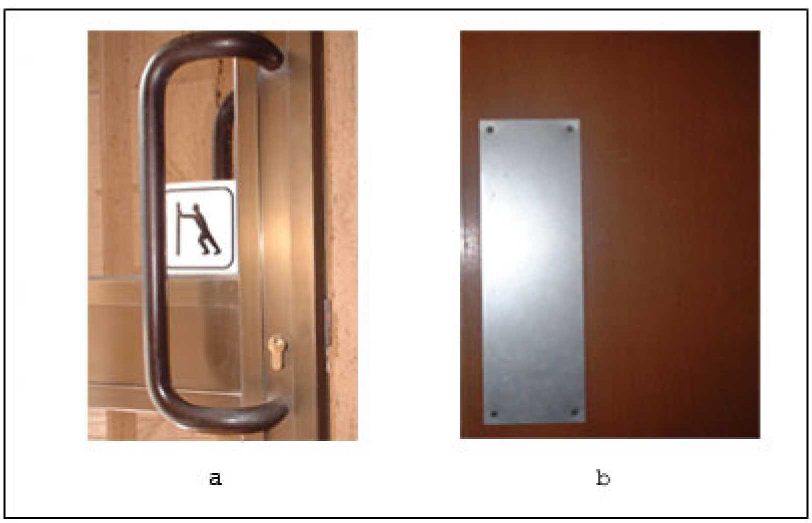

Figure 1 Two door handles, one (a) very confusing as to its usage, and one (b) which gives clues as to its usage.

To explain the difference between a physical and a perceived affordance, consider the following example. Consider the door, which opens when pushed with a flat plate that takes the place of the door handle (Fig. 1b). The design of the door handle gives out the clue that the door is not supposed to be pulled, since there is no handle that the actor can grab in order to pull the door. Conversely, a door handle that can be grabbed (Fig. 1a) gives out the clue that the door opens when pulled. However, as Norman (Norman, 1988) points out, this convention is not always followed, resulting to people thinking that they cannot figure out how to open a door, whereas the problem lies in bad design, and bad use of a perceived affordance. In fact for the door in question (Fig. 1a), someone has placed a sticker to

show that the action that opens the door is the opposite of the action that the door handle suggests. This is a very good example of using a perceived affordance to convey the wrong information.
The difference between the real affordance, or the affordance as defined by Gibson, and the perceived affordance in Norman's definition is that the door affords to be opened in some way (Gibson's definition), but the perceived affordance that the flat panel gives out is that the door can be opened by pushing on the panel (Norman's definition).

2.1.2 Constraints

Constraints are mentioned in many places in the literature as giving helpful clues to the user of an interaction artifact, about how the artifact may and may not be used (Norman, 2003). Constraints in general are attributes of artifacts that limit the way the artifacts can be used. Consider the example of an elevator button. When pushing the button, it can only be pressed so far. That constraint allows the user to realize that he/she has completed the pressing action, since the full range of the pushing of the button has been reached. Thus now the user knows that he/she has to release the button. In this way, the constraint that limits the range of forward/backward motion of a button, also gives a very specific clue to the user; namely that the pushing action is completed. Thus in the limitation that this constraint provides for the button, it also gives a hint to the user as to how the button is used.
Viewed in this way, constraints are a way that a designer may use to provide knowledge in the world, knowledge that may guide the user to perform an action only in the boundaries that are acceptable for the designed interface. Constraints do not have to be only physical, like the one in the previous example. Norman (Norman, 2003) mentions many different kinds of constraints, like physical, cognitive and cultural.
Cultural constraints are constraints that culture imposes to people. For example, red means stop or danger in many western cultures. The stop lights of all cars are red, and most people in the world realize that when a car's stop lights are turned on, it means that the car is decelerating, which means that they, in turn, need to decelerate too, or face some kind of danger. This is a constraint that is imposed by culture or rules. Thus, a warning that is accompanied with some kind of red color will be understood faster as a warning, than if it was accompanied with a green color.

2.2 Task Knowledge Structures

Task Knowledge Structures (Johnson and Johnson, 1988, 1991a, 1991b) is a theoretical and methodological approach to modeling tasks. The theory assumes that when people learn how to perform a task, the actions and methods used to perform that task are not stored in memory as stand-alone facts. Rather they are stored as grouped knowledge structures that can be remembered and recalled as a coherent whole. This knowledge about tasks is represented in conceptual structures in long term memory, as a coherent structure for each task, called task knowledge structures (TKS). It is assumed by the theory that all of the knowledge about some task is stored in a TKS, which can be activated in association to task performance (Johnson and Johnson, 1991a).
TKS include knowledge not only about how to perform a task, but also about objects that are used to accomplish that task. This knowledge allows people to recognize these objects and use them according to the requirements of the task and according to the people's experience with these objects. Knowledge about these objects then includes knowledge about their affordances and constraints, mentioned in the previous section.
TKS were created as a tool to aid design generation. By modeling user knowledge and using the associated methodology, a designer can use the theory to generate design solutions for interactive systems (Hamilton, Johnson and Johnson, 1998). This is in contrast to other task analysis methodologies like GOMS (Card, Moran and Newell, 1983), which is purely an evaluative theoretical model.
TKS have two overriding principles: taxonomic categorization and sequential dependency. The taxonomic categorization principle states that objects that are similar are grouped together, and that users create higher level declarative knowledge structures that allow them to remember actions on the group rather than on specific objects. The sequential dependency principle is based on the assumption that if people are to carry a task, after they perform the first action that will bring them closer to the completion of a task, they will not go through every possible action that they know in order to find the next best action to take. Rather, especially if they have performed the task before, they will have a group of actions stored as a sequence which when performed, will bring them to the desired result. This dependency between the actions and the sequence that they have to be performed in, is the principle of sequential dependency. Like taxonomic groupings, actions are grouped together in a meaningful sequence that allows for the creation of subgoals that will lead to the completion of the task (goal).

These principles were tested and evidence was provided to show that they are indeed correct (Hamilton, Johnson and Johnson, 1998). Even though Hamilton et al. (Hamilton, Johnson and Johnson, 1998) talk about objects in TKS and hint at the affordances of objects, the object roles are not explicitly defined in terms of a user interface, nor are the affordances and constraints of these objects included in TKS. An extension of TKS is proposed, that includes these two items of information so that the identifiability principle will work by using the theoretical basis of TKS.

3 Definitions

The comparison of any interaction style to another can only be accomplished if there is a common vocabulary that researchers and designers can use to describe the different parts of the interface. In the following sections we propose terms that may be used to create this common language needed. The terms presented each describe a different part of an interface with respect to what it represents in a user interface.

3.1 Data Objects

Every interaction style has some sort of representation for the actual data that resides in the overall system. Even though all data is described by 1's and 0's in the end, each interaction style has some sort of representation of the logical groupings of that data, be that graphical, physical, textual or otherwise. The representations are used to represent distinct data entities such as files, folders, database tables, applications, etc.

A data object is defined as the representation of a distinct data entity or group of entities in an interaction style. Data Objects may take many different forms and may be used under different guises, but their sole purpose is to allow the manipulation of the data entities that they represent by the user. Data objects may be comprised of other data objects; they can be combined to create more complex data objects, or may include other data objects.

Data objects have attributes that fall under two categories. Descriptive attributes, which are used to describe the data object, and usability attributes that are the attributes that define the ways that data objects can be used in an interaction.

Descriptive attributes of data objects are the ones that allow users to perceive the data objects as such. These attributes include sensory affordances, which were mentioned in the previous chapter. Color, weight, and size are also descriptive attributes. Usability attributes are the attributes that allow the user to perceive the functionality of the data object. These attributes have more to do with the affordances and the constraints that are designed into the object. For example, the writing on a button that describes its functionality would be a usability attribute of the data object. It would not be a descriptive attribute, because it doesn't describe the data object. Rather, it describes the function that the data object performs.

3.2 Interaction Objects

In any interaction style there are things that allow the user to interact with the data objects. The way the interaction is carried out between the user and the data object takes many forms. For example, the user may use the pointer to drag and drop a file from one folder to another. In this example interaction, the user uses the pointer (the interaction object) to manipulate the file (the data object).

Interaction objects are the objects that are perceived by the user to be the means of interacting with the data objects in an interaction style. They are the tools that the interaction style provides to the users that allow the manipulation of properties, states and attributes of the data objects. Interaction objects don't have to be manipulated directly by the users, although this is possible.

An interaction object may manipulate one or more properties, states and attributes of the data objects. Also an interaction object may act on one ore more data objects.

Interaction objects are used when the user cannot directly control or manipulate the data objects, or when the user cannot directly control or manipulate specific properties or attributes of the data objects. Interaction objects may also be used in the case when the interface wants to explicate to the user a specific way of manipulation, or an interaction with an attribute of a data object.

3.3 Intermediary Objects

Intermediary objects are the actual objects that are used to manipulate the interaction objects. Intermediary objects are usually physical artifacts in any interaction style. These artifacts are used to manipulate the actual interaction

objects in an interaction style, when the interaction objects cannot be manipulated directly by the user. An example of an intermediary object in the direct manipulation interaction style is the mouse. The mouse is the device with which the user interacts with the pointer, is an intermediary object because it is the conduit, the real world object by which the user gives commands to the pointer in order to be executed in the interface.
Intermediary objects add a level of complexity to any interaction style because the user must not only learn to use the interface, but also learn to use the intermediaries and how they control objects in the actual interface.

4 Identifiability

An idea was presented of how different interaction styles could be put on an equal footing (Christou and Jacob, 2003). The idea was based on the principle of comparing interaction styles on how many things the user would need to learn in order to use these interaction styles effectively. The intuition behind this idea was that if the user needs to learn fewer things in order to use effectively an interaction style, then that interaction style would be “better”, in the sense that the user would not have to expend too much effort in order to learn how to use the tool to perform a task, but rather would be able to perform the task almost from the start.
The “things that the user would need to know” that the aforementioned authors mentioned, are the TKS that the user needs to have in order to perform a specific task in a specific interface. The idea though, extends further than that. TKS can be broken down into showing the knowledge that a user needs to have in order to perform an action that will bring him/her closer to the intended goal. Problems though occur when the user either has incomplete knowledge about the performance of all the actions needed in order to complete the task, or when the user cannot remember how to perform a specific action in order to complete the sequence that will bring about the completion of the task.
Identifiability provides a measure for the knowledge that is available for the user to find in the world. This knowledge can be used to guide the user to complete an action when the user does not know or does not remember how to perform an action in the sequence.
Norman (Norman, 2003) talks about visibility, and how things should be made clear to the user in respect to the conceptual model, the results of actions, etc. The authors understand this to mean that the IO, IN or DO should allow the user to understand their approximate use.
Another way to view identifiability is that it is a measure of the knowledge in the world. The interface style should give to the user the necessary clues, so that s/he would be able to learn much about the interface, just by looking at its tools, and by working with it. Identifiability is based on the Appropriateness principle [1] which states that: “The representation used by the artifact should provide exactly the information acceptable to the task: neither more nor less”. Essentially, the identifiability measure does not take into account any kind of information that the artifact presents to the user. Rather, it only takes into account the information that the artifact presents to the user that is relevant to the task that the artifact is supposed to accomplish. In this way, one can view identifiability as the measure of the perceived affordances under Norman’s definition [7] or as the cognitive affordances of the IO, DO, or IN, using the term cognitive affordance under Hartson’s [5] definition, that are relevant to the task at hand.
Notice that the user mostly utilizes knowledge in the world only when s/he doesn’t have any knowledge in the head [3, 4]. Identifiability becomes a factor when the user knows something about the interface style, but also needs a reminder about something that s/he does not use as often.
Identifiability is the sum of the attributes that an IO or IN have that allow the user to recognize or infer their usage. It is knowledge in the world that is given to the user in the form of affordances and constraints. In this chapter we will explain the concept of identifiability. We will also elucidate why it is important to be able to measure the identifiability of an interaction design or artifact and propose a way of measuring identifiability. Finally, we will show how the concept can be used to change an IO or IN to make it more useable.
Identifiability as a measure is the count of perceived affordances and constraints that an object has that are relevant to the task. Identifiability is based on the Appropriateness principle (Card, Mackinlay and Robertson, 1991) which states that: “The representation used by the artifact should provide exactly the information acceptable to the task: neither more nor less” (Card, Mackinlay and Robertson, 1991). Thus essentially, the identifiability measure does not take into account every piece of information that the artifact presents to the user. Rather, it only takes into account the information that the artifact presents to the user that is relevant to the task that the artifact is supposed to help the user accomplish. In this way, one can view identifiability as the measure of affordances of and constraints on the artifact that are relevant to the task.

Since it is hard to find the affordances that do play a role and convey information for some task, 3 questions are proposed that may help the designer and researcher to identify which affordances may be taken into account for this measure.

1. Which are the perceived affordances and constraints of the artifact?
2. Which tasks does the artifact permit the user perform?
3. Which perceived affordances and constraints convey information to the user for each action?

Taking into account these questions will enable the researcher to identify the affordances and constraints that the artifact has which pertain to a specific task, and use them to get a count for the identifiability measure.

5 Experiment

An experiment was devised to assess whether the identifiability measure is a good predictive and evaluative measure. The experiment consisted of having a group of nine (9) experienced MS Windows XP™ users perform a file move task. The subjects were all taken from the computer science department of Cyprus College, and their age varied from 19 to 25 years old.

5.1 Method

The task was designed on two different interaction styles: one condition was based on MS Windows XP™, which represented the direct manipulation interaction style, and the second condition was designed on a tangible user interface, by using a mockup of the actual interface. Since the interaction with the interface is what is of interest here, it was deemed unnecessary to actually build the tangible user interface to support the task. No information was given to the subjects as to how to perform the operations that the experiment called for.
The task was first converted into a TKS. The procedure, as delineated by Johnson and Johnson (Johnson and Johnson, 1991a) was not completely followed, because the actions and the task were predetermined for the experiment. The role that the subjects were asked to enact was of a user who wishes to categorize their mp3 (song) collection, which is a generic role that many users of college age usually perform. The specific task that the subjects were asked to perform was the following: "Find a specific (given by the experimenter) mp3 file in a list of folders, where the source and target folders are given, and place it in the correct folder, according to the performing artist". The object that the subjects were asked to act upon was an mp3 file in a specific folder. The "mp3 file" object has the properties indicated in table 1, which uses the extended taxonomic substructure categorization to include the affordances and constraints of the object. The extended categorization also includes the type of the object in the interface (data, interaction, intermediary).

Table 1 Task category of "mp3 file" object in the Direct Manipulation Interaction Style

Category	Information
Is a … of the interface	Data Object
Is a member of	Move file
Is used in procedures	Categorize files according to some attribute
Is related to … by …	Drag and drop as target object
Is associated with actions	Drag and drop, move, cut and paste
Is manipulated using interaction object	Pointer
Has features	Filename
Typical instance	An mp3 file which has an audio file icon
Affordances	1. Known icon type by user. 2. Filename reveals content.
Constraints	1. Must be manipulated with pointer in specific way known to user to perform move.

The subjects' initial view in the direct manipulation interface condition is shown in figure 2.

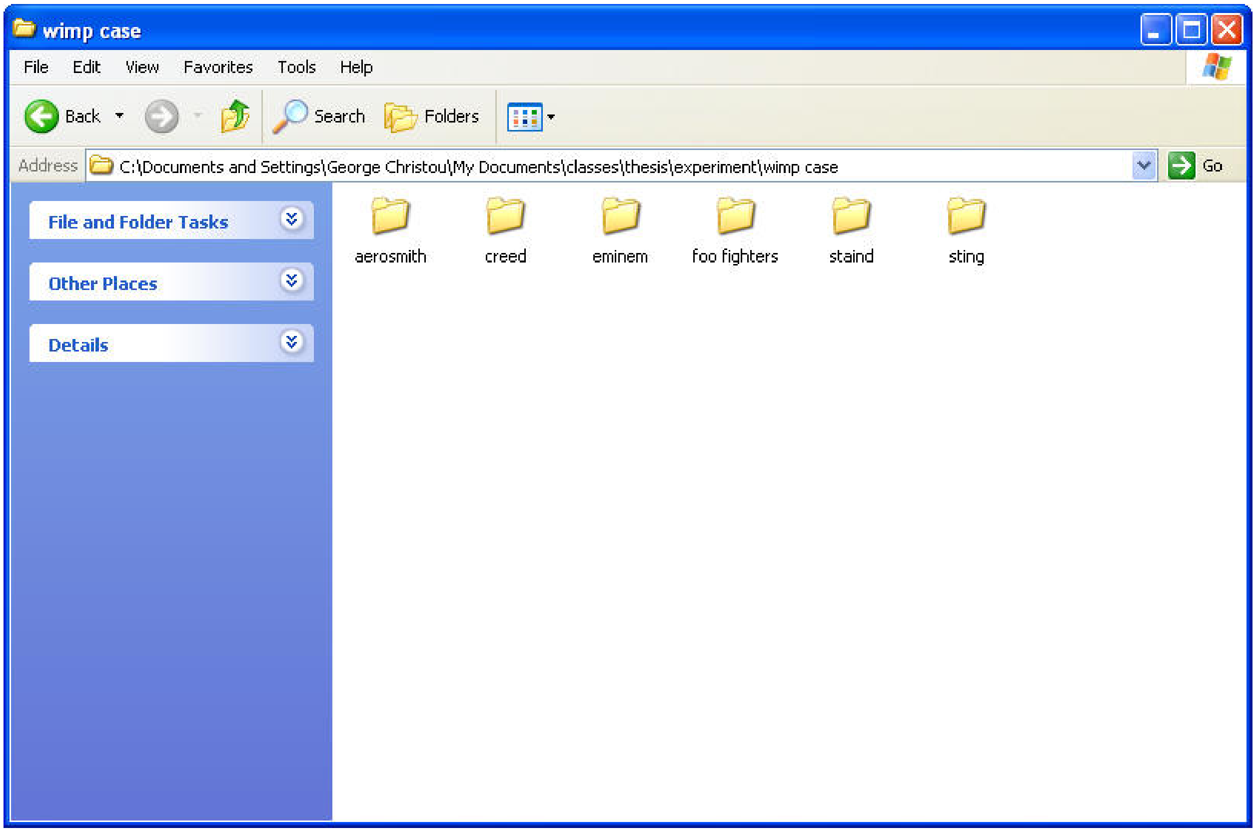

Figure 1 Initial view of direct manipulation condition

Table 2 represents the extended substructure categorization of the “mp3 file” object in the TUI interaction style.

Table 2 Task category of "mp3 file" object in the TUI Interaction Style

Category	Information
Is a … of the interface	Data Object
Is a member of	Move file
Is used in procedures	Categorize files according to some attribute
Is related to … by …	Grab, put into as target object
Is associated with actions	Move
Is manipulated using interaction object	None
Has features	Filename
Typical instance	An mp3 file represented by a phicon (Calvillo et al, 2002)
Affordances	1. Tangible object that can be manipulated with user’s hands. 2. Filename reveals content. 3. Manipulated exactly like objects in real world.
Constraints	1. Phicons (Calvillo et al, 2002) cannot exist outside of the containers (cognitive constraint).

The subjects’ initial view of the Tangible User Interface is illustrated in figure 3.

Figure 2 The Tangible User Interface case

The identifiability measure provides the scores shown in table x for the two conditions of the task.

Condition	**TUI**	**Direct Manipulation**
Score	3 affordances, 1 constraint	2 affordances, 1 constraint

Table 3 Identifiability Scores for each Condition

According to the identifiability measure, the subjects should perform better in the TUI condition rather than in the Direct Manipulation condition, because of more affordances of the data objects in that condition.

5.2 Questionnaire Results

The most significant result from the identification point of view is that when users were asked in the pre-experiment questionnaire whether they recognized what the boxes and cards inside the boxes were, they all answered correctly. All the users correctly recognized the boxes as containers for the mp3 files, and the cards inside as the mp3 files relevant to the container. As mentioned previously, since no instructions were given to the subjects as to how to perform the task in the two conditions, it means that the subjects were all able to use knowledge they already had from the direct manipulation interface and real world interactions and apply them to the task at hand. The subjects were also asked how they would perform the delete action, and again every one answered correctly, but the action was not necessary for the completion of the experiment.

5.3 Experiment Results

This section presents the results of the second run of the experiment, using ANOVA analysis to show that the difference in performance time is indeed statistically significant, and is presented in table 3. The performance times for each subject are shown in table 4. The subjects were the same that participated in the questionnaire part, whose results were presented in the previous section. This time though, they were asked to actually perform the task, in both conditions (in the TUI interaction style and in the direct manipulation interaction style). The presentation of the conditions was determined by a coin toss, to randomize the effects that the order of presentation may have had on the subjects. The results of the statistical analysis clearly show, with statistical significance of $p<.005$ that the subjects performed much faster in the TUI condition than in the direct manipulation condition.

Table 4 The ANOVA analysis

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	364.2301	1	364.2301	10.31959	0.005434	4.493998
Within Groups	564.72	16	35.295			
Total	928.9501	17				

Table 5 The times for each subject for the experiment

	WIMP	TUI
Subject 1	8.00	1.80
Subject 2	7.5	5.53
Subject 3	13.35	4.03
Subject 4	12.33	2.36
Subject 5	7.37	4.02
Subject 6	21.33	6.07
Subject 7	31.22	8.05
Subject 8	7.86	2.24
Subject 9	9.12	3.01

Thus the result that the identifiability measure provided is indeed correct and coincides with the experimental result.

6 Conclusions

This paper presented the concept of Identifiability, which is a comparative, evaluative and predictive measure for tasks designed in different interaction styles. TKS where presented and were extended to be used in taking measurements for identifiability, by adding affordances and constraints for each object that was categorized and used in the task represented.

A common nomenclature was presented which allows for the discussion of the various parts of user interfaces, and various attributes where defined for these parts of user interfaces, according to their function. Namely, data objects, interaction objects and intermediary objects where defined and it was shown how they can be identified and used in the measurement for an interface.

Finally, an experiment was presented that provides evidence towards the usefulness and validity of the identifiability measure.

References

Beaudouin-Lafon, M. (2000). Instrumental Interaction: An Interaction Model for Designing Post-WIMP User Interfaces. Proc. ACM CHI'00 Human Factors in Computing Systems Conference.

Card, S. K., Mackinlay, J. D., and Robertson, G. G. (1991). A Morphological Analysis of the Design Spaces of Input Devices. ACM Transactions on Information Systems, Vol. 9(2).

Card, S., Moran, T. and Newell, A. (1983). The Psychology of Human Computer Interaction. , Hillside, NJ: Lawrence Erlbaum Associates.

Christou, G. and Jacob, R. J. K. (2003). Evaluating and Comparing Interaction Styles. In Jorge, Joaquim; Jardim Nunes, Nuno; Falcao e Cunha, Joao (Eds.) Interactive Systems. Design, Specification, and Verification 10th International Workshop, DSV-IS 2003, Revised Papers Series: Lecture Notes in Computer Science, Vol. 2844, XIII, Springer-Verlag.

Gaver, W. W. (1991). Technology Affordances. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. New York: ACM Press, 79-84.

Gray, W. D., Fu, W. (2001). Ignoring Perfect Knowledge In-The-World for Imperfect Knowledge In-The-Head: Implications of Rational Analysis for Interface Design. CHI Letters, 3(1), 112-119.

Gibson, J. J. (1977). The Theory of Affordances. In R. E. Shaw and J. Bransford (Eds.) *Perceiving, Acting, and Knowing*. Hillsdale, NJ: Lawrence Erlbaum and Associates.

Gibson, J. J. (1979). *The Ecological Approach to Visual Perception*. Boston, MA: Houghton Mifflin Co.

Gray, W. D., Fu, W. (2002). Soft Constraints in Interactive Behavior: The Case of Ignoring Perfect Knowledge in the World for Imperfect Knowledge in the Head. Cognitive Science

Hamilton, F., Johnson, P., and Johnson, H. (1998) Task-Related Principles for User Interface Design. Van der Veer, G.C. (Ed). Task Analysis 1988 – State of the Art in European Approaches, Schaerding, June 2-4.

Hartson, R. (2003). Cognitive, Physical, Sensory and Functional Affordances in Interaction Design. *Behaviour & Information Technology*. *22*(5), 315-338.

Hutchins, E., Hollan, J., & Norman, D. (1986). Direct Manipulation Interfaces. In D. A. Norman & S. W. Draper (Eds.) User Centered System Design: New Perspectives in Human-Computer Interaction. Lawrence Erlbaum Associates: Hillsdale, NJ.

Jacob, R. J. K. (1995). Natural Dialogue in Modes Other Than Natural Language. Beun, R. J., Baker, M. and Reiner, M. (Eds.) Dialogue and Instruction. Berlin, Germany: Springer Verlag. pp. 289-301.

Johnson, P., and Johnson, H. (1991a) Knowledge Analysis of Tasks: Task Analysis and Specification for Human-Computer Interaction. Downton, A. (Ed.) Engineering the Human-Computer Interface. London, U.K.: McGraw-Hill. pp. 117-144.

Johnson, H., and Johnson, P. (1991b) Task Knowledge Structures: Psychological Basis and Integration into System Design. Acta Psychologica, 78 pp. 3-26.

Johnson, P., Johnson, H., Waddington, R., and Shouls, A. (1988) Task Related Knowledge Structures: Analysis, Modeling and Application. Jones, D.M. and Winder, R. (Eds). People and Computers IV, Manchester, Cambridge University Press. pp. 35-62.

McGrenere, J. & Ho, W. (2000). Affordances: Clarifying and Evolving a Concept. *Proceedings of Graphics Interface 2000*. Montreal

Norman, D. A. (2003). The Design of Everyday Things. New York: Basic Books.

Shneiderman, B. (1983). Direct Manipulation: A Step Beyond Programming Languages. IEEE Computer, 16(8).

Shneiderman, B. (1998). Designing the User Interface: Strategies for Effective Human Computer Interaction. Addison Wesley

Van Dam, A. (1997). Post-WIMP User Interfaces. Communications of the ACM. February 1997, Vol. 40(2).