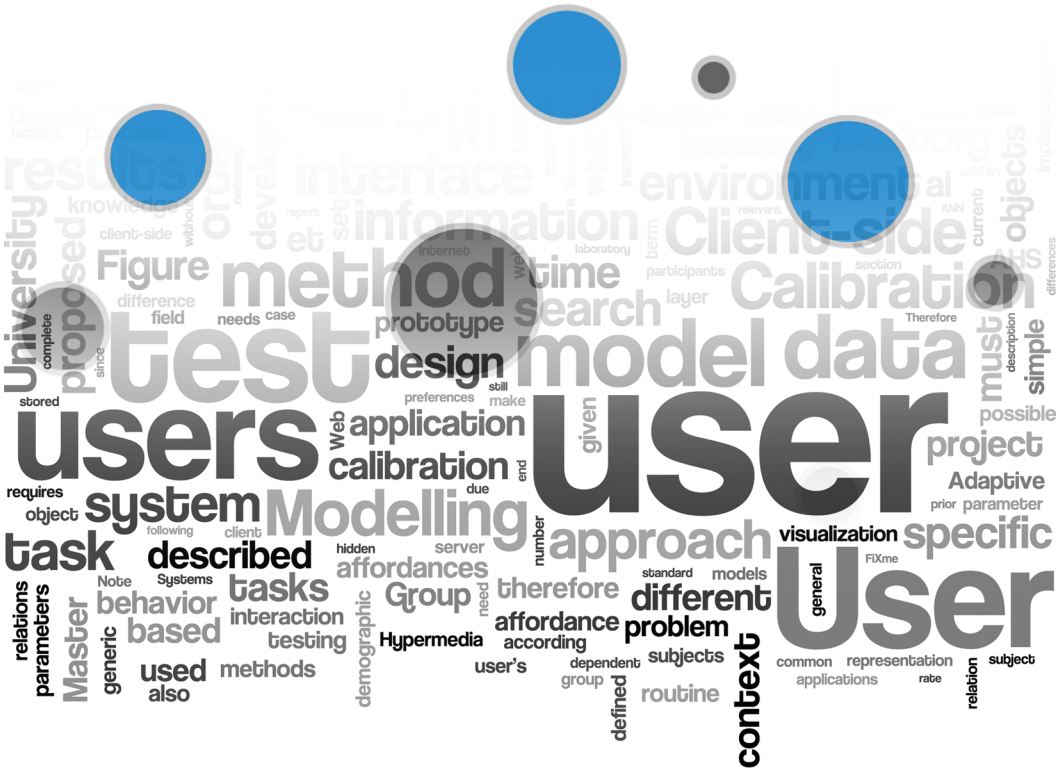
Master Thesis 2011

# Generic detection of user interaction behavior




Christoffer Baadsgaard
Thomas Wisbech Pedersen
Medialogy - Group 111031

Supervisor: Hans Jørgen Andersen
Department of Architecture, Design & Media Technology

AALBORG UNIVERSITY
STUDENTS STUDYREPORT

AALBORG UNIVERSITY
STUDENTS STUDYREPORT

**Department of Media Technology**
Niels Jernes Vej 12-14
Phone +45 9940 8792
Fax +45 9940 9788
http://www.imi.aau.dk

**Title:**

Generic detection of user interaction behavior

**Description:**

Investigating user behavior & ability in interface design, using quantifiable methods

**Theme:**

Master thesis

**Project period:**

10th semester:
February 7. 2011 - Oct 31. 2011

**Participants:**

Thomas Wisbech Pedersen

(Technology Specialization)

Christoffer Baadsgaard

(Design Specialization)

**Supervisor:**

Hans Jørgen Andersen

**Publications:** 6

**Number of Pages:** 125

**Finished:** Oct 2011

**Study:**

Medialogy - Media Technology & Design

**Abstract:**

Increasing data spaces, complex & relation-based data structures, and individual customized data is challenging the current data representations, mainly limited to a list-based approach, which does not describe the relationships between data objects. When data structures are changing towards semantic markup, the relations between data objects become valuable and can be stated as *meta-affordances*, which the user can utilize in understanding relationships between data.

Adaptive systems that relies on User Models have previously been based on user history or models of the complete environment. Open environments, such as the Internet, requires a different approach & this project provides a method to generically evaluate differences in user behavior that could refine existing User Models. Data from a simple visual indexation tasks that the user solves in a set of different environments, serves as a basis for detecting user behavior. The results are that detection of user behavior using simple generic calibration is possible in visual interfaces - the method could also be applied in HCI in general which will be encouraged by the authors.

# Preface

## Formalities

Sources are referred to by ["author's surname", "year the text is written"], the literature list is drawn up with the authors' surnames in alphabetical order.

A video presentation of the problem is available on the DVD in the Appendix C along with a PDF-version of this report, a demo of the high fidelity prototype from the test, and PDF-versions of the cited sources.

## Acknowledgments

The project is based on open source frameworks and applications - thanks to the community putting the time and effort into publishing all this software for free. The sheer and overwhelming amount of work out there prove the exciting drive people have toward creating new ways of interfacing our growing information-base.

We would like to thank our advisor Hans Jørgen Andersen for his patience. We would also like to thank Erik Granum, Morten Lund, and Betty Li Meldgaard for their valuable feedback to both the report and the test. Finally, we would like to thank all the participants in our test for taking the time to complete the tasks, which made the investigations in this thesis possible.

The work is dedicated to everyone who has financed our studies & friends and family who have accepted our absence while being commited to our studies and this master thesis.

*In loving memory of Ruth Klausen.*

# Contents

# 1. Analysis

## 1.1 Motivation

Information tecnology has reached a level where the digitization of almost all accessible information has created an enormous amount of digital data. One of the most significant signs of that development, is the Internet, which can be seen as a cornerstone in the information society and a reflection of the current tendencies in information technology, where the digitization of many processes causes the stream of information and data to be rapidly increasing. Data and information can be easily accessed anytime and everywhere and many traditional activities are now done online, e.g. a lot of the communication between the State of Denmark and the citizens is performed digitally. Along with data being accessible by the Internet users, an increasing number of contribution sources is adding to the globally available knowledge base. The data stream basically consist of statically submitted data, both published and user-generated, and dynamically generated data, which makes the Internet an infinite data space, since data from e.g. search queries and other dynamic sources only exists when a user or a computer agent makes the query. In order to combine the data into information, from which the user can subtract knowledge, as visualized in Figure 1.1, the data must be processed by a computer system and presented through an interface, like e.g. a search engine or a computer application. The link between the user and the data space is therefore the various interfaces that process the data into comprehensible information.

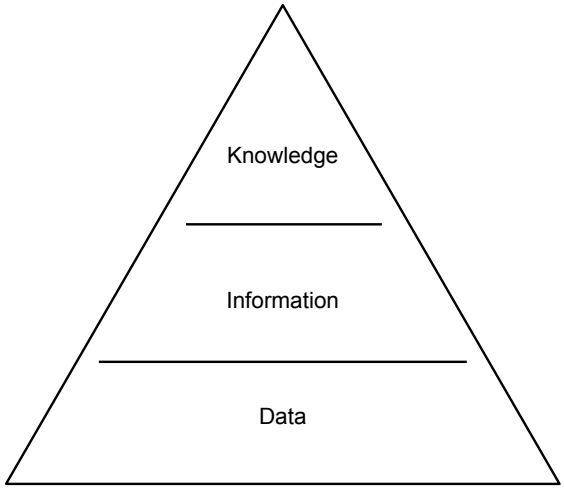



**Figure 1.1:** The issue for interface developers is to transform data into knowledge for the users. This is becoming a more acute issue as the amount of data increases exponentially.

Because the data is becoming increasingly complex and large in amount, the content presented to the users, is tailored to fit the individual user's preferences, based on submitted information, e.g. a user profile and history like previous search queries. That adaptation is made rather statically on the server-side before the information is presented to the

user. While the content is filtered on the search engine server, the representation of it is made in the user's client. Accessing a vast amount of data or information, like browsing on the internet, is currently confined to a very limited visual representation, where most search engines use a list based approach to show their results to the user, regardless of the data structure. Though, not all kind of data is suitable for existing representation and the process from transforming complex data over information to knowledge, sets the requirement of a proper representation. Currently, information representation is limited to a list-based approach, which creates the paradox:

*Content is individual but the representation is general.*

While the amount of data is increasing, furthermore, the current trends in the overall structure of the Internet is towards more relation-based data, which will be described in Section 1.3. The Semantic Web is focused on the correlations between data in contrary to the current ranking, where link popularity is the dominant factor and therefore new kind of data representations might occur; when visually describing complex relation structures, a simple list based on popularity is not sufficient. The Semantic Web will be used in this project as an example of a complex data layer from which a given interface needs to establish an information layer which the user can utilize for obtaining knowledge. The content is tailored to the user by using a profile, the interface should also be adaptive to fit the user's needs. When presenting information to a group of users, there is a risk that the level of information is either too low, which will limit the individual user, or too high which will cause "information overload". However, a suitable representation is dependent on many factors; many of the factors are dependent on the current state of the user and the surrounding environment, which therefore cannot be based entirely on history alone, but requires calibration to a model of the user's state, as visualized in Figure 1.2.

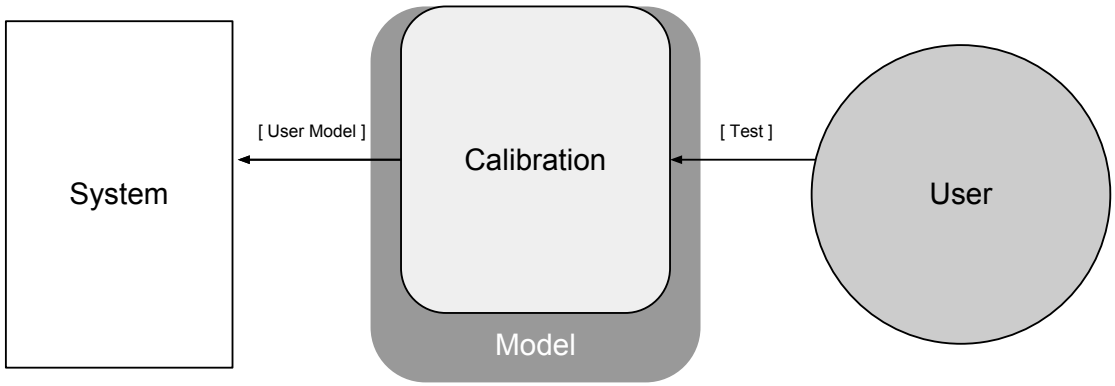



**Figure 1.2:** The project seeks to investigate if simple user behavior can be segmented in order to establish a simple calibration routine for a *User model*, using a generic simple approach.

The methods found in the research of this project all involve complicated models of both the users and the context of their environment, hardware, etc. In contrary, this project seeks to find a more simple approach, by investigating if the users' behavior in an interface can reveal measurable differences, which can be utilized for detection of the users behavior. There are no silver bullet to adaptive user interfaces and it is still in the software industry primarily a design issue and not a user preference & adaptation issue. This calls for alternative visualization methods which cope with complex data without compromising the user experience. The design of interfaces with alternative representation of data requires knowledge about how users cope with digital information and the limits on how much data they can utilize.

An approach to measure and fit the digital models to the users and their environments is Adaptive Hypermedia Systems (AHS) that benchmarks the individual user's performance and calibrates the system to present data in accordance to that, which will be described in Section 1.5. The principles of AHS will be used in this thesis as a theoretical method for calibrating an interface to the user, by creating a dynamic *User model* which represents the user's current state. The focus is narrowed down to a specific part of the *User model* - investigations of users' behavior in a client-side browser interface and whether or not their patterns of interactions can by measured and used to determine the individual user's current behavior.

This thesis is primarily focused on the method of evaluation and the derived results, rather than the implementation of a specific interface itself. This is due to a generic consideration of making findings more applicable in various contexts rather than being tied to a specific context. The evaluation method is remote and asynchronous thereby removing the need for real-time evaluation by test facilitators. The test approach involves users solving one simple task in different visual scenarios, with 4 different types of distraction to detect the behavior of users & whether or not it is possible to detect differences in the interaction behavior to determine the users current method of interaction, or internal state. This experiment will in a broader perspective be used to verify whether a generic calibration routine for user models could be a valid approach. Interesting findings and useful perspectives in such as any results which can be used for segmenting users by their behavior, is sought through analysis of recorded user behavior in a confined interface. If any measurable differences can be detected, they will be analyzed and used as foundation for adding behavior to a user model, or use such measures as a description of the usability of an interface.

## 1.2 Complex Data Structures - the Semantic Web

The communication between machines and humans can be divided into 3 different layers, *Data*, *Information*, and *Knowledge*, as shown in Figure 1.4. The former, *Data*, does not provide anything useful to the user before being processed into Information by some kind of interface. As described in the Motivation, Section 1.1, the amount of data is increasing rapidly and the structure of the Internet, which serves as the primary example throughout the Analysis chapter, is furthermore becoming more complex. In order to fully understand the need for the investigations made in this thesis, an introduction to the Semantic Web is necessary.

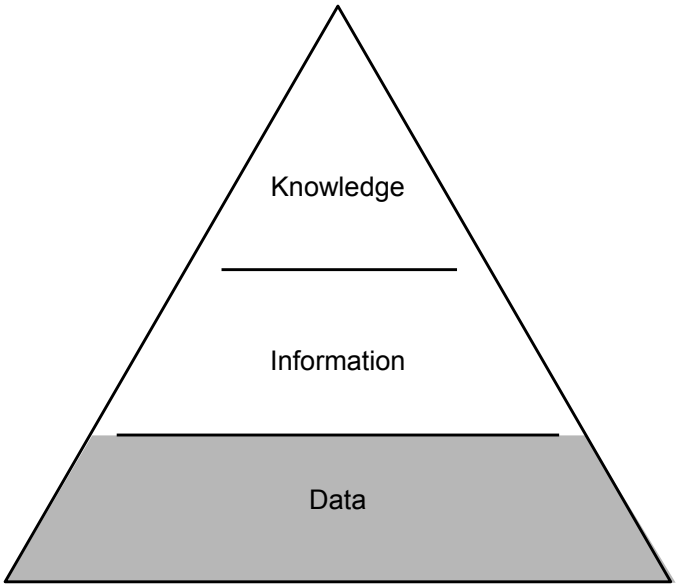



**Figure 1.3:** The *Data* layer is typically computer generated and often incomprehensible to users. This section describes the data layer behind Web 3.0, i.e. the Semantic Web.

After the wave of web applications designed with focus on information sharing, user-centered design, and collaboration emerged on the internet, which was coined as the term, "Web 2.0", by Tim O'Reilly at a conference in 2004 [Allen, 2009, p. 1], the Internet has become a rapidly growing information repository, where only the most popular data survive through the search engines from the social networking sites, blogs, wikis, video sharing sites, hosted services, web applications, and mashups. With various tools and applications, providing virtually anybody with capabilities of using a computer, the opportunity to publicly submit data, the term, "Web 2.0", can be seen as a reaction to the rise and fall of the "Dot Com Bubble", which bursted in the start of the millennium; a range of technology with massive investments, but no content caused an urge for user submitted content.

A new term, "Web 3.0", has marked the next paradigm of the Internet as a technology. While the third generation of the Internet has many definitions from various entities, the

World Wide Web Consortium (W3C) has the most prominent, stating that a substantial part of Web 3.0 will be the Semantic Web. The W3C founder, Sir Tim Berners-Lee, defined the Semantic Web in his initial visions for the Internet:

> *The Semantic Web goal is to be a unifying system that will (like the Web for human communication) be as un-restraining as possible so that the complexity of reality can be described.* [Berners-Lee & Fischetti, 1999]

To describe the *complexity of reality*, as Tim Berners-Lee mentions, descriptions are defined as an *ontology*, which is a description of features of data objects, their environment and their relations. Each data type in an object must have a defined *ontology* in order to achieve a unified standard of comparison between objects. The *ontology* of a specific application of an object, is inherited from a modularization of a generic *upper ontology*, describing the overall environment, in relation to a *domain ontology*, describing the subject area, and a *task ontology*, describing a specific task within the *domain ontology*, visualized in Figure 1.5. These ontology descriptions are formalized in Resource Description Framework (RDF), which is the foundation of the semantic data structures.

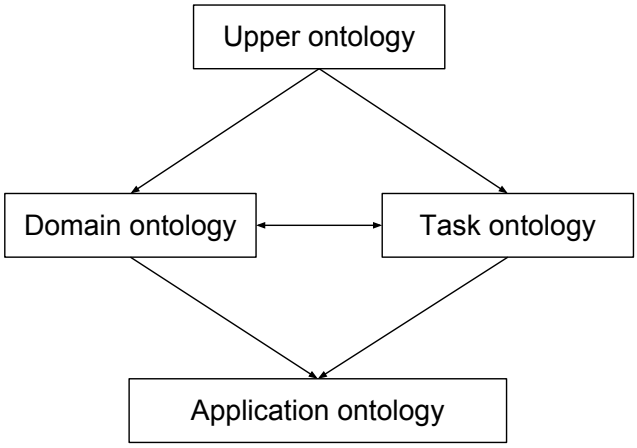



**Figure 1.4:** The modularization of ontologies based on scope and partial ordering of inheritance [Obitko, 2011a]

The Semantic Web is still in development as a concept, including descriptions of the technologies involved, in the realization of accessing the Internet as an ontology of interconnected data structures with complete descriptions of the data and its interrelations, but has improved from being a theoretical vision to a future technology worth mentioning in the media [Andersen, 2011]. The complexity of the relations for a search query for the word, "Gene", is visualized in Figure 1.6.

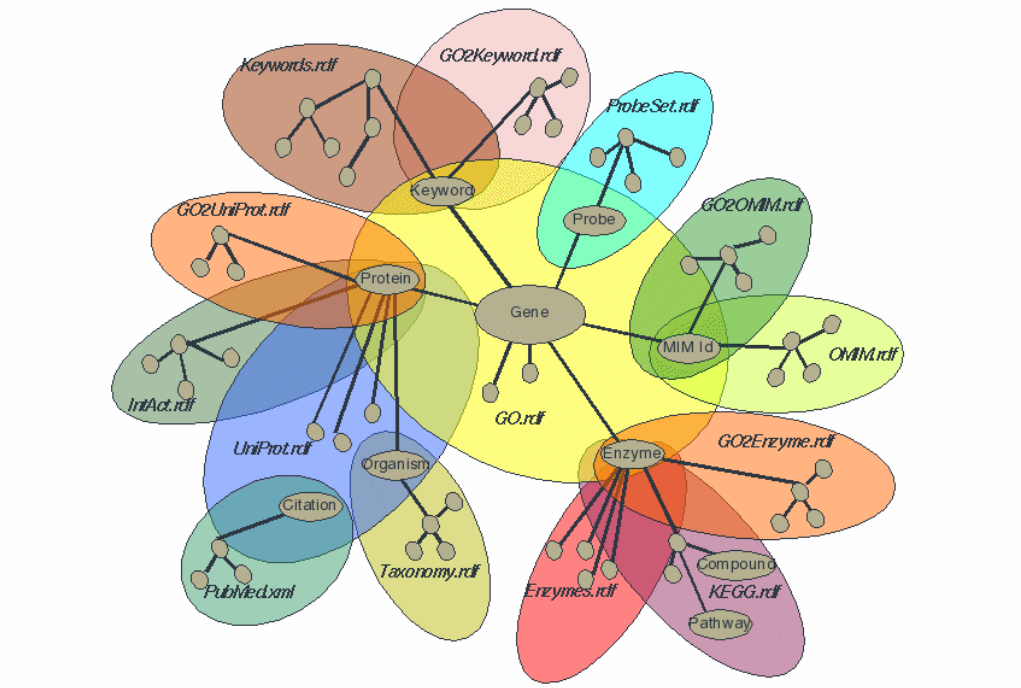



**Figure 1.5:** A visualization example of the relations between a search subject, "Gene" and its various correlated subjects [W3C, 2006]

The concept of the Semantic Web, visualized in Figure 1.7, is described by a stack of architectural layers, connecting the *user interface* to the specific *URI* of the resource. While some of the layers are based on existing technology (marked with **bold** on the figure), others rely on theoretical concepts not yet implemented. In order to control the validity of the data, control layers are placed between the user interface and the underlying layers. First layer is *Trust*, where the user is authorized, typically through login or digital signatures, which ensures cryptographic security for the actual data. Second layer is *Proof*, which is a control layer for proving the validity of the received inputs in order to verify the data relations. The *Unifying logic* layer is a generic method of computer-reasoning, based on inference of the received information and general logic principles to crawl the Internet resources and semantically markup the data and the corresponding meta data. This has yet to be formalized and implemented in practice. One approach to semantic markup of meta data is by using RDF-schemes (RDFS), which consist of triples of *subject-predicate-object*, describing the object. *Querying* are made by SPARQL, which is a query language capable of performing queries of triple patterns, which is suitable for the RDF(S)-format. The *Data interchange* is made with RDF and the *Syntax* is XML. Finally, the *Identifiers* are traditional URIs known from the existing Internet structure and the *Characters set* is Unicode, which is a characters set that contains all characters from languages around the

world.

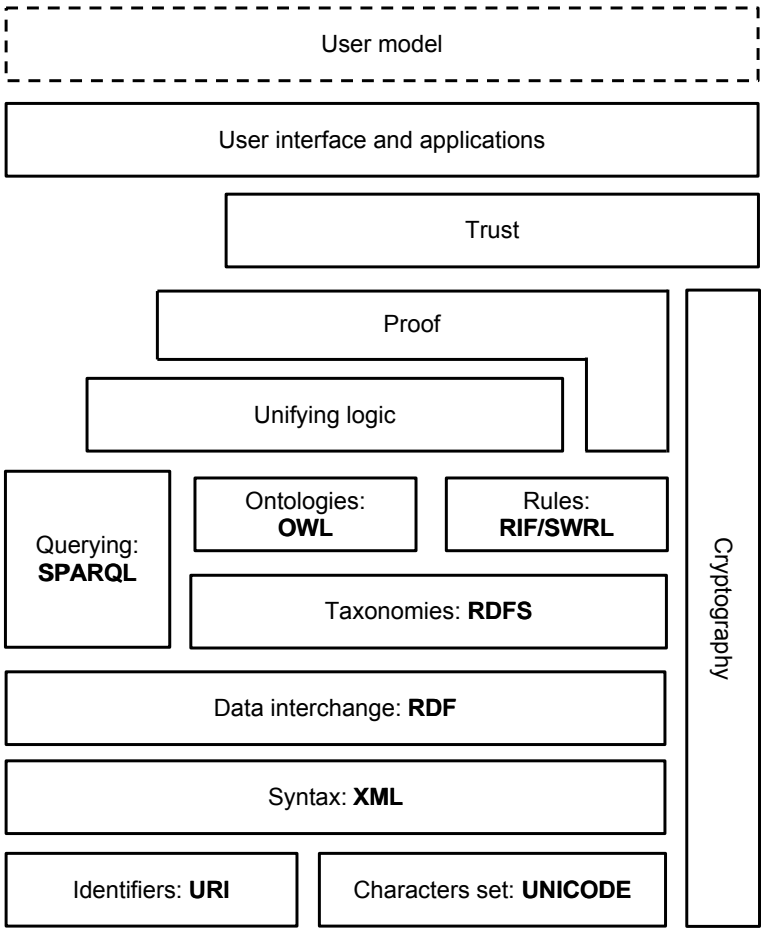



**Figure 1.6:** The architecture layers of the Semantic Web. [Obitko, 2011b] (modified). The text in **bold** represents already implemented technologies, while the other layers have not yet been formalized in terms of a uniform standard. The figure is modified with the addition of the *User model* investigated in this report.

The extensive *ontology* of the relations between objects produces a massive amount of data to be processed when searching in the semantic search space. The paradigm of the Internet changing towards a more thorough and semantic data representation where ontologies of the relations between objects are becoming a substantial part of the object's ranking. A long with a rapidly increasing rate of data contributions, the enormous amount of data requires a unifying method for representation of data, in order to digitally portray *the complexity of reality.* The principles of the Semantic Web is summarized in the Semantic Web Stack, which consist of several architectural layers, required for accessing specific data with consideration to the semantic description and the relation to other data. As ontologies are used for description of data objects, their features, environment and relations,

a dynamic and unified standard of comparison between objects is needed in order to cope with an infinite types of objects and related feature sets. An adaptive data representation by utilizing a *User model*, as visualized in Figure 1.7, is therefore needed in order to fit the to the user.

## 1.3 Connections between data - *Meta-affordances*

The previous section described the structure of the Semantic Web where the amount of data is increasing dramatically because descriptions are made of all data relations within the existing *ontologies* of the web. The data, without being processed, does not tell the user anything, but being formalized into comprehensible information, the user has the possibility to utilize them. An interface or application can process raw data into information which is understandable by the users. To the user, these ontologies can be considered as a set of *affordances*, where the connections between the data and their description, of a given object is in that regard an *affordance* to the user.

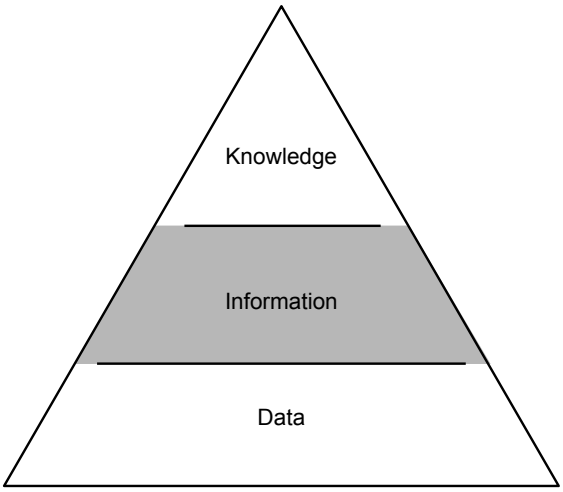



**Figure 1.7:** The information layer is a metalayer between the data and knowledge where connections between data becomes a description of the relations between data. This *meta-affordance* will be discussed in this section along with the original affordance design principle.

*Affordances* is a popularized way of describing users' feelings and connotations about objects and the term has got a broad prospect of interpretation. It was originally introduced by psychologist, James J. Gibson, who defined affordances as all the objectively measurable possibilities of actions, dependent on the user's capabilities, in a given environment without consideration to the his culture, prior knowledge or expectations [Shaw *et al.*, 1977]. Since the introduction of the term, many interpretations of the term has been coined, differing from the original meaning. Affordances in the context of this project is based on the User Interface perspective described by Donald A. Norman in his book "The Design of Everyday Things" originally from 1988 [Norman, 2002]. In this book, he describes physical objects in relation to their potential usage as affordances. An affordance in the context of a physical object is the implicit knowledge that human beings have of, for example a ball - it is round, it can bounce, deflate, roll, which is independent of the user's knowledge about it

and considered its *actual affordances* by Gibson and furthermore it can be used in sports like football, handball, etc. which is dependent on the individual user's knowledge and considered as *perceived affordances* by Norman. These affordances can be either known to the individual handling the ball or unknown, which can be either due to its specific design or the individual's lack of knowledge about the potential usage. As a real world example, soccer ball manufacturers tries, by ballistic testing, to design their balls for optimal floatation which will affect the users perception of the affordance that describes how the ball handles in the air. These properties would also be considered *perceived affordances* to the users' who is able to recognize them. Norman's interpretation of affordances differs from Gibson's by classifying all subjective feelings from the user towards an object, as affordances regardless of their mapping to actual functions. Gibson objectively classifies all properties as affordances, whether they are recognized by the user or not. Norman would classify affordances given the user's ability to perceive an object & its properties.

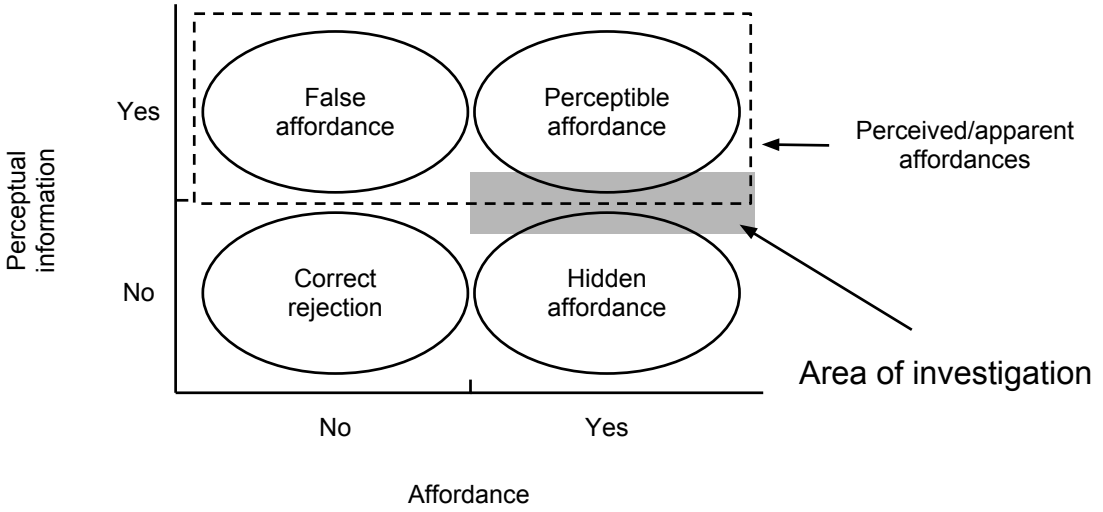



**Figure 1.8:** Mads Soegaard has formalized Donald A. Norman's and J. Gibsons's affordance term of an attribute in a HCI context in order to identify whether an attribute has the appropriate affordance to the user [Soegaard, 2010]. The model is modified to emphasize the area of investigation that is the threshold when users find an affordance to be either hidden or perceptable. Norman classifies affordances, whether or not it actually provides the property

The difference in the two interpretations is shown in Figure 1.9, where affordances can be mapped according to the perceived information; if the object design "promises" a certain property without providing actually it, it is considered as a *false affordance*. In contrary, a property hidden to the user is considered a *hidden affordance*. A *correct rejection* is a property which is correctly classified by the user to be irrelevant of the intended use.

In an Interaction Design context there is primarily a focus of determining whether the

user / focus group can use a given layout - whether it has the relevant information for the affordances in the interface to make sense. Determining the limitations of an affordance in a given application is a different approach that could potentially tell something about both users and the interface they are using.

The casual use of the word, *affordance*, has since its conception made it a buzzword without the potency of its original intent of describing a *physical* objects where the affordances of the object are implicitly useful and perceptually obvious to the user. Donald A. Norman argues against the use of affordances as a concept that applies to digital interfaces which must be learned. As an alternative, he proposes to use a combination of constraints and feedback to accommodate and test for the affordances - that according to him only applies to physical objects [Sharp *et al.*, 2007, p. 33].

For this project we use the term, *meta-affordances*, as a method of describing the ontological information about the features of virtual objects. In the Semantic Web, all objects are described by ontologies of their relations to other objects, as described in Section 1.3. A *meta-affordance* is when the property of an object is determined by its relations to other objects; if the user do not recognize the connection between the objects by utilizing the meta-data (object descriptions, relations, etc.), the *meta-affordance* is considered hidden. In contrary, information shown to the user, which does not benefit in completing the intended task, is considered as a false *meta-affordance* and useful information which the user does not recognize is considered a hidden *meta-affordance*. Therefore, the classification of the state of the user and these *meta-affordances* is dependent on investigations on how much information users are capable of perceiving when interacting with an interface.

An example of a *meta-affordance* is an object's relation to another object in the results of a search query. A simple search query for the word, *"soccer"*, will result in relations to various topics, which might be more or less relevant to the user, according to the user model, as shown in the simplified search tree in Figure 1.10.

The illustrated user in Figure 1.10 proves to be a (most likely male) sport fan and when searching for the term, *soccer*, visualized relations to *sport* in general and further down to other sports like football with *aussie rules* and *motocross*, are within his interests in relation to the search query and therefore become perceptible affordances, while *handball* and the non-sports relations clustered around *hooliganism* have no interest in relation to the search and should be classified as correct rejections, even though he finds *hooliganism* related to *racism*, *alcoholism*, and *crime*.

> *Proper customization does not come by further complicating an already complex system. No, proper customization comes about through combining multiple simple pieces. Invariably, if something is so complex that it requires the addition of multiple "preferences" or customization choices, it is probably too complex to use, too complex to be saved.* [Norman, 2005]

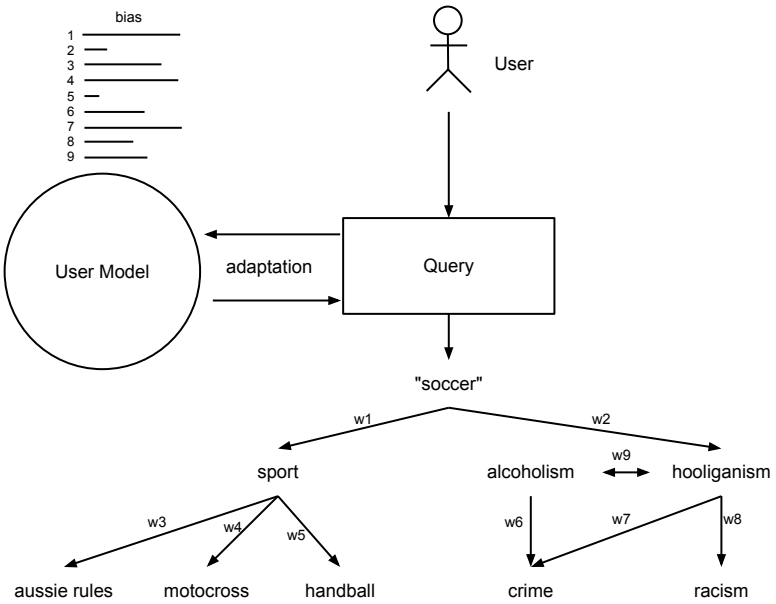



**Figure 1.9:** When a user search for a term, the relations between search results become meta-affordances. The meta-affordances are determined by the search query in relation to the user model, which weigh the relations to the user's preferences, search history and environment / context

According to Norman's quote, too much configuration makes the product too complex for the user so the adjustment between the presented data and the user can not be done by making the user fill out several preferences, but on the other hand an interface without restrictions would present too much information to the user who would be unable to utilize it. Therefore, an interface between the data layer and the information layer could be beneficial by being adaptive so it automatically fits the user's capabilities & behavior, without being dependent on advanced configuration by the user - as seen in Figure 1.7.

## 1.4 Knowledge Distribution to Users - Adaptation

The final layer in the communication between computer and human is the knowledge layer, which is dependent on the individual user's ability to utilize the presented information. If this ability could in any way be quantified by the system a more custom and relevant representation could be delivered to the user.

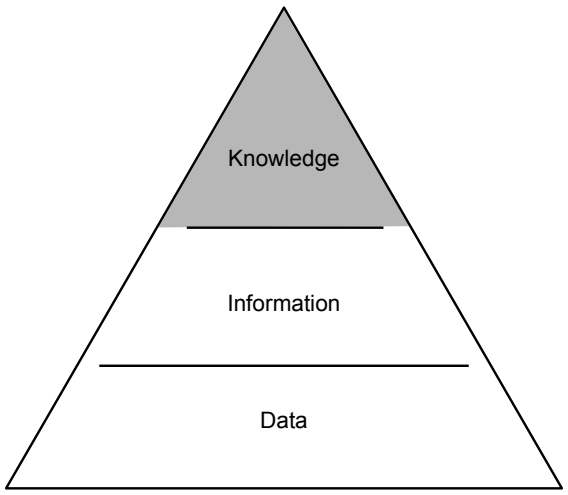



**Figure 1.10:** The knowledge layer describes systems that communicates the information from the lower layers to the users. In this section Adaptive Hypermedia Systems (AHS) will be described as an adaption method between the information layer and the user who subtracts knowledge from the presented information.

Adaptive Hypermedia Systems (AHS) is a scientific umbrella for research in dynamic and adaptive interfaces. The field is in a riveting development at the moment, because of an increasing need to develop efficient methods for aiding users navigating successfully in continually increasing amounts of information. Other systems, such as Adaptive Educational Hypermedia Systems (AEHS), provides information according to the user-profile and their current estimated knowledge base, and only exposes the users to content that fits their current abilities. The AEHS have in common that they have a limited *Domain model* as the full knowledge base, typically the curriculum of a given class. In order to understand Adaptive Hypermedia as a term, a brief introduction will follow and subsequently there will be a discussion of the latest discoveries and approaches that have been published in the research field.

Adaptive Hypermedia Systems (AHS) is working with digital models of users, the environment they operate in and an adaptation model that work as the feedback mechanism between the two models. The quantification of users, the environment, etc., is usually problem specific in the sense of the purpose of an application or approach, and will therefore vary a lot in between applications.

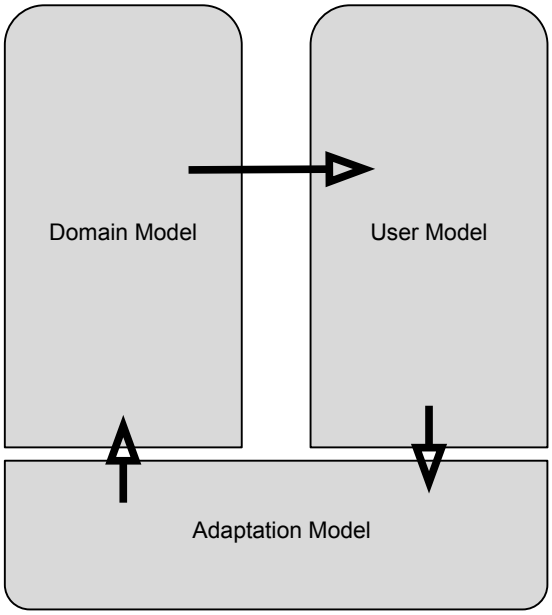



**Figure 1.11:** Adaptive media uses 3 models to make adaptation to the user possible. A Domain Model representing the environment, the User Model representing the user and the Adaptation Model as the feedback layer.

- *Domain Model* - a representation of the environment that users navigate through in a given application. The environment is quantified into variables or features that digitally represent that environment, which both can be physical and completely digital.

- *User model* - a representation of a specific user. The user's preferences, abilities, etc., are also quantified into the model. The *User model* is dependent on the overall purpose of a given application and is also a design decision as with the *Domain model*.

- *Adaptation model* - different to both the *Domain model* and the *User model*; it is a feedback layer controlling the output to the user, taking into account the parameters of both models. It modifies the accessibility of the *Domain model* using the parameters in the *User model*. This adaptation to the *Domain model* is what makes the Adaptive Hypermedia approach work as an intermediary between the users and the domain they operate in.

The origins of AHS is traced by citation back to [Brusilovsky, 1996], where Brusilovski coins the term and tries to make a general approach to it. The usual use and research of AHS has been based on server side processing. Specifically in regard to *User modeling*, there is extensive use of the information that the server receives of the user input such as user history, personal data, etc. This information has been used to create the *User model*

on actions that involve the server only. Retrieving only server side information for the *User model* does not give a full picture of the user's abilities in a given interface. Working with the user in this way, the system is only capable of detecting when the user is successful in regard to interaction with the interface; nothing is stored on the server, if the user's operation fails due to non-server issues. This is an important point because server side communication is in a sense the derivative of successful interaction.

> *Traditionally, user modeling in Adaptive Hypermedia Systems is based upon monitoring requests of resources on the server. This information, however, is relatively vague and may lead to ambiguous or uncertain conclusions. One approach to improve granularity and accuracy in user models is monitoring client-side user interactions in the browser. Despite their steadily growing importance, especially along with the emergence of Web 2.0 paradigms, up to now such interactions are hardly being monitored.* [Hauger *et al.*, 2011, p. 147]

The other less obvious problem than server side observations of users is that websites become monolithic [Hauger *et al.*, 2011, p. 147], in the sense that the gathered data does not tell where the user interacted in order to successfully submit requests to the server. This can of course be inferred, but the the behavior leading up to interaction with the server is not representable in a server side observation context.

### 1.4.1 Current trends & latest findings in AHS

The User Modeling, Adaption and Personalization 19th International Conference, UMAP 2011 - conference, held this summer in Girona, is the latest conference held on the subject of Adaptive Hypermedia Systems. A series of approaches are described in the proceedings, which will be handpicked to fit this thesis; the methods and proposals from the proceedings will be described and used in accordance to be useful in the research aims of this project. The proceedings shows that the latest research implies it can be useful to make client-side observations of the individual user to get better insights into what the users are doing, while interacting with an interface. The following findings by various authors all relate to the topic of performing client-side investigations of the users' behavior in an application interface, which is important in relation to the topic of this thesis.

The findings of [Hauger *et al.*, 2011, p. 152] documents by both tracing eye-movement and mouse movement of users on a webpage that *"The mean distance of mouse and eye position reduces to less than 50% when users are clicking, selecting text, or when the mouse is moving."* The methods all try to segment the former monolithic websites into smaller generic areas, which are not defined by the elements in the page rather an area of the page. The findings of [Hauger, 2011], show an asynchronous approach implemented to *User modeling* in Adaptive Educational Hypermedia Systems, where learning styles and

other former implicit approaches are modeled with success. The proposed approach relies on the events triggered by interaction with the interface, and not by server submission.

This goes hand in hand with findings of [Parra & Amatriain, 2011], who finds a correlation between explicit and implicit feedback that users give to a system, in this specific case on the webpage *"last.fm"*, that services users with music. The findings are, that there is a correlation between the amount of time spent on listening to a piece of music and the rating it gets by the user [Parra & Amatriain, 2011, p. 266]. This is probably obvious to most people, but it has interesting implications that *User models* can quantifiably be representative of the likes and dislikes of a given user.

When talking about the likes and dislikes of the user, there is also in the proceedings a team that focuses on the *Activity Recognition* of the user which finds that using events instead of regular intervals. When determining user activity, [Ortiz Laguna *et al.*, 2011], reduces the amount of clutter in the data, and improves the prediction model of the users' behavior.

Adaptive Hypermedia Systems are composed of a parametric representation of a *User model* describing the users, a *Domain model* that describes the environment, where the users navigate, and an *Adaptive model* that governs the interaction between the *Domain* and *User model* and acts as the feedback mechanism between the two. The ability to adapt a *Domain* to a *User model*, proves to be of great value in Adaptive Educational Hypermedia Systems, where the users are only exposed to information that fits their current abilities. These systems operate within a limited *Domain* and can therefore be tailored to a very specific knowledge domain.

The current trends in AHS shows an increased emphasis on tracking users as they interact with the client instead of with the server, which has been used to track the users history, etc. Because the users are tracked as they interact, the *User models* can be dynamically changed by monitoring activity. The thorough exposition of the theories behind, examples of actual implementations, and the latest findings shows that the principles of AHS is suitable for adapting a complex data representation to a *User model* representing the user's abilities.

## 1.5 Adaptive representation in systems

The design of adaptive representation of interfaces can utilize the principles from AHS, where the presented information in a given system is a weighted result between three models, representing respectively the User, the Domain, and Adaption. The latter being a link between the User and Domain, which works as a controller for the feedback from the system and can be seen as the essential part of AHS, see Figure 1.12. AHS is using a *Domain model*, which is only a representation of and not the complete *ontology* of the working environment, which can be very costly in terms of machine power and can be incomputable and assumptious at best. Observations of user behavior could limit the need for a total representation of the working environment and the user - because deviances from usual behavior would indicate that a necessary *meta-affordance* is currently hidden to the user.

An example of implementation could be in a common applications such as the word processing program, Microsoft Word, which most people have encountered through either work or studies. The program is a WYSIWYG text-editor, capable of a broad spectered branch of options from text formatting, typography, and spell checking to writing invoices, light Desktop Publishing (DTP) assignments, and even simple webdesign. Because of its status as a multipurpose tool for everything related to working with text, the interface must be able present all possible options to the user according to the task at hand. The developers have chosen to implement a "Home" tab ("Startside" on the figure, as it is the Danish version), where presented options as a design choice that the development team have assumed that their users would need the most are listed, as shown in Figure 2.2.

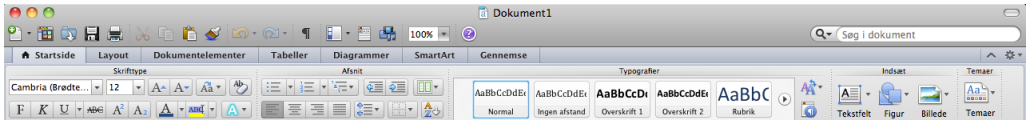



**Figure 1.12:** The standard toolbar of Microsoft Word 2011 (Mac edition, Danish version) shows a broad range of tools, which the developers have prioritized as important to most users.

In order for the users to make their own toolbar if the standard set of preferred options does not fit the purpose of their particular needs, they will have to add a new toolbar from a menu and then drag every desired option into the newly created toolbar from a selection list of more than 1.000 options. This process is very cumbersome and fits well to Donald A. Norman's quote about customization, that "*... if something is so complex that it requires the addition of multiple "preferences" or customization choices, it is probably too complex to use, too complex to be saved*" see Citation 1.4. Instead, the customization could be made by a method using AHS for adaption of the interface to the specific user as a generic calibration routine, which detects the user performance by observing the user's

interaction with the system against an optimal usage pattern. A Microsoft Word user who often uses the mathematical functions and symbols, should have her interface dynamically calibrated so that the most used functions would be most significant and be listed so that they are immediately available in the interface. On the contrary, a user who writes Spanish essays should have a dynamic list containing signs such as "¿", "´", "‘", etc., which are specific for the Spanish language. This is somewhat accomplished in the spell-checker that analyses the language that the user composes their text in, but the application GUI itself is not changed to fit the current usage. Thus a dynamic interface could solve the affordance issue by evaluating itself against the *Domain* and *User model*. The critical aspect would be that the user who regularly uses functions in relation to a specific type of task would have difficulties occasionally performing other tasks, since the system would be calibrated to another usage pattern. Therefore, the system must be able to take this into account by allowing re-calibration at all times. A re-calibration can be done in several ways, but 3 obvious options seem suitable for this specific example:

1. A *User model* for each cluster of detected use patterns, e.g. "Math", "Spanish".
2. An entropic function, which will deviate gradually from the established *User model* to introduce the users to new affordances.
3. A complete re-calibration by resetting the current *User model*

These options can be seen both as individual solutions and as supplements to each other. Option 1 will require that the user labels the pattern of each use session, as visualized in Figure 2.3, when closing down the program. In this way, the user will have to save the recorded session as e.g. "Math" when finishing a mathematics report. This way, the "Math" setup can be chosen at startup or be detected by the system according to the behavior, to decide the profile, the next time the user needs the mathematical functions. If the user wants to write a letter, an option for a new, blank session, which subsequently can be saved as e.g. "Letter, private" will occur. Option 2 would be a function with 2 possibilities - "Less functions" and "More functions", adding more or removing affordances to the particular user profile. The weights can then be adjusted through calibration or preference setting so that the *Adaption model* can dynamically adjust the interface to the *User model*, and algorithmically decide to select the available affordances, also visualized in Figure 2.3. This functionality could easily supplement the first option, expanding or constraining the various profiles depending on the user's requirements, while only introducing one parameter to the user such as an entropy parameter to describe how rigid the dynamic profile should be. The 3. option would simply be to reset the recorded user behavior for either a specific profile or for the whole system and re-calibrate it according to new user patterns.

In opposition to the example above, searching an unstructured search space as the Internet, requires the ability to sort data according to a *User model* or server-side *User profile*

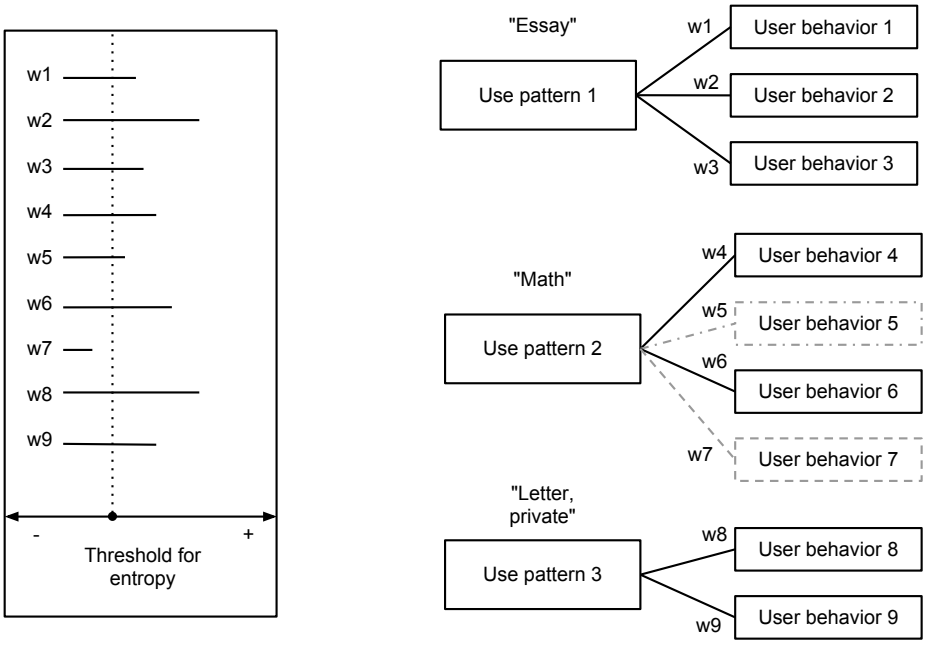



**Figure 1.13:** 3 set of use patterns clustered from user behavior, subsequently labelled by the user as "Essay", "Math", and "Letter, private" (Option 1). The threshold for entropy have excluded "User behavior 7", because its weight (w7) is too small. By adjusting the threshold, more or less entropy, the user is able to include w7, if the setting is too narrow or to exclude other weights (next would be w5), if the setting is too broad (Option 2). In this illustration, the threshold is defined for all weights together, but could be implemented for the individual use patterns

as is common practice in modern search engines regarding the content. The introduction of client-side calibration of the *User model* would give more precise descriptions of the users, their current device, behavior and needs. Not only would a dynamic calibration be able to describe the users and their needs more specifically, it would also be able to detect the state of the user if it fits a pattern. This refined prediction could provide both users and service providers with benefits, like Google who are in the business of trying to predict what their users want. If this classification of behavior is to be made in accordance with a client-side *User model* along with a server-side *User profile*, a more refined model would emerge that describes the user behavior with a given interface in contrary to only using server-based a *User profile*, as visualized in Figure 2.4. This information could determine whether the *meta-affordances* in the interface are hidden or perceived to the user.

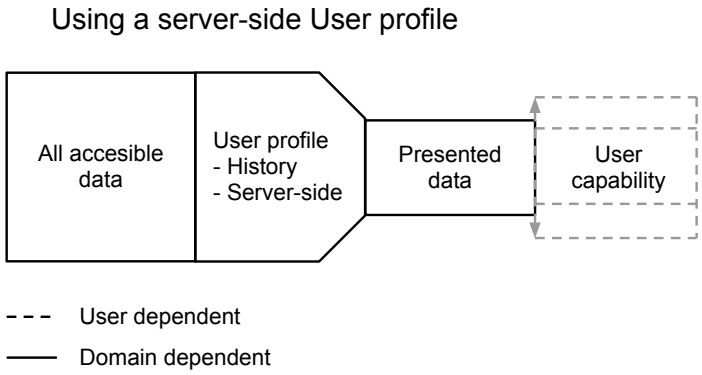



**Figure 1.14:** A *User profile* based on predefined server-side assumptions of the user, might differ from what the user is actually capable of perceiving.

The client-side *User model* will, by dynamic calibration, adjust and adapt to the real-time measured state of the user, who therefore only will receive the information, which she is capable of perceiving, as visualized in Figure 2.5. Because of that, the presented information can serve as *meta-affordances* to the user.

The classification by *User model* is performed client-side and the *meta-affordances* are determined by how many entries of the returned search result the user is capable of perceiving and utilizing. A search for the term "web" will return various results with different relations to the query, as visualized in Figure 2.6. The top shows all the returned results in relation to the query. The *User model* of the specific user might limit the numbers of returned search results as a result of the user's behavior and only return some high weighted results, shown in the middle of the figure. From these weighted results, the user is able to navigate further and center the relation ontology to another term, in this case "html", as shown in the bottom of the figure. If the returned results do not satisfy the user, there is the possibility to perform another search or broaden the results space. As with the

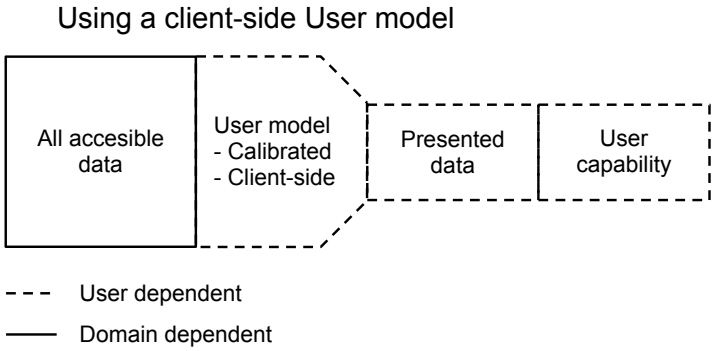



**Figure 1.15:** A *User model* based on realtime client-side calibration of the user, will adapt to the current mental state of the user and thereby only present the information actually perceivable by the user.

previous example, the user should be able to define by an entropy function how much each parameter should be weighted, allowing the results to vary more away from the *User model*.

Recent research have shown that it is possible to detect users' skills by only observing their interactions, classifying them into respectively novice or skilled users [Ghazarian & Noorhosseini, 201 A method describes that it is possible to detect the skill level of users by analyzing the lengths of pauses in interaction with a user interface achieved through chunk identification [Santos & Badre, 1994].

> *A main characteristic of skilled behavior is smooth and continuous operation. In order to operate smoothly, users should have fewer pauses with shorter durations.* [Ghazarian & Noorhosseini, 2010, p.124]

The ideal implementation of a calibrated client-side User Model would therefore need to be able to detect certain aspects of the user behavior in order to adapt the given interface to that specific user. Though, the described current methods are dependent on modeling by complex classifiers, which predict that users will improve their skills over time. Assuming that users are linearly improving, such predefined methods are missing that users at a certain skill level might perform differently dependent on factors in their mental state or working environment, and the purpose of the task. A generic calibration routine for client-side user modeling could facilitate such changes by being recalibrated by the user if they deem that the conditions have changed radically. Furthermore the application could suggest recalibration of the system if there is too much deviance between current and calibrated behavior.

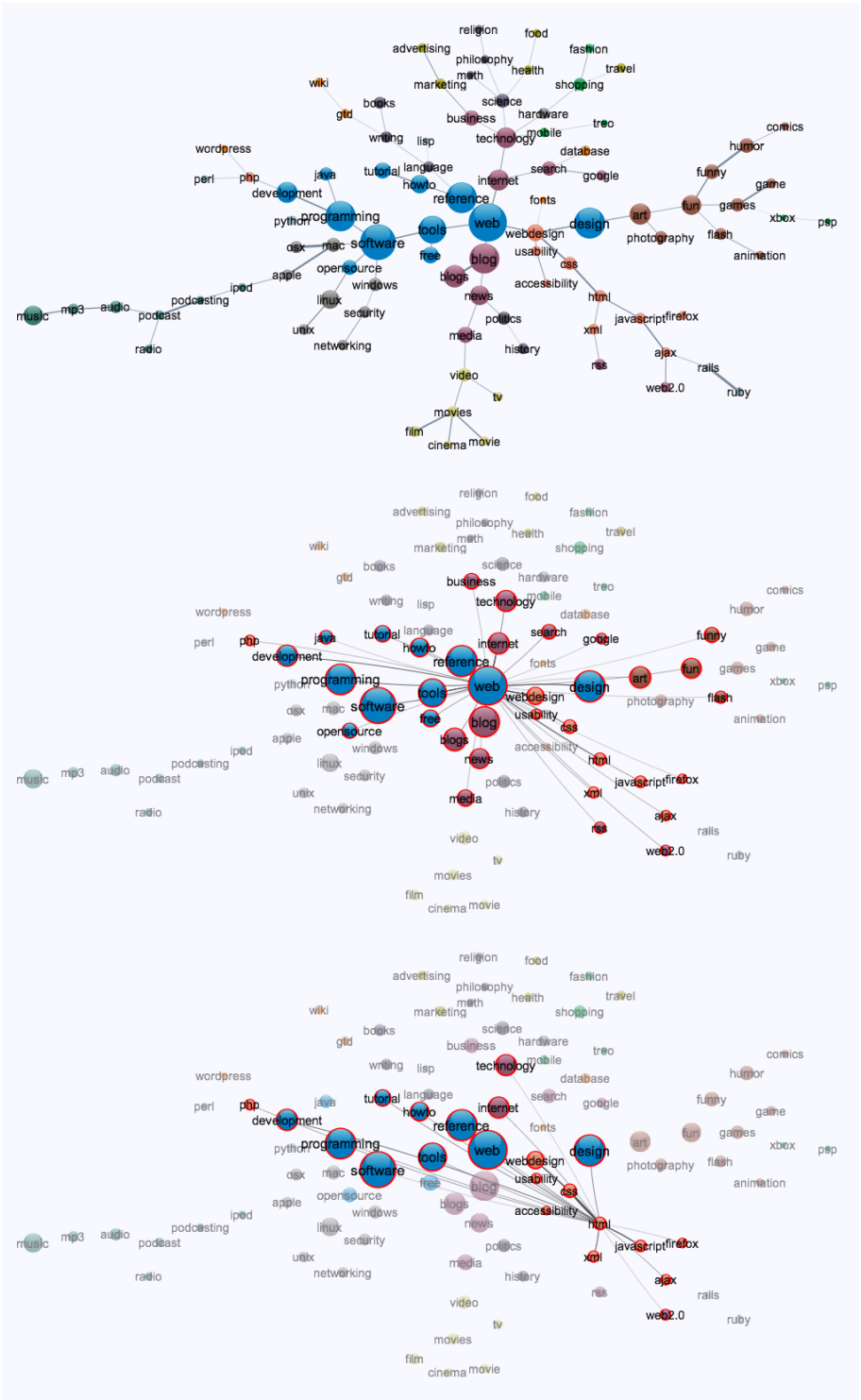



**Figure 1.16:** An illustration of a search for the term, "web", would create clusters of search results with the potential of being *meta-affordances* to the user (top). A calibrated *User model* will limit the number of results according to the user (middle). From the weighted results, the user can proceed the search by choosing e.g. "html", which will center the search focus and thereby the relations on that term (bottom). [Montero & Solana, 2007]

### Artificial Neural Networks & AI

When examining Figure 1.12 more closely, there is a resemblance between that and an Artificial Neural Network (ANN). The feedback mechanism of the *Adaptive model* reveals itself to be of the same functionality as the hidden layer in ANN, which consists of a minimum of 3 layers: An input layer, a hidden layer, and an output layer. ANN & Artificial Intelligence (AI) is of great interest in the development of AHS, because they are adaptable. This section will describe a few examples of how it has been implemented in earlier research projects, and how they differ from each other.

AI in learning requires that the *Domain model* is completely defined and that it knows its actions a priori, in order to function as intended. [Papanikolaou *et al.*, 2000, p. 629] show that the two concepts are actually contradictory, instead of complementing each other, because the AI approach needs to be highly structured & have a complete *ontology* of the problem field prior to deployment, and therefore making most of the design decisions in the design and development process instead of during the run-time process, where AHS are supposed to operate. The ANN approach is a lot looser in its coupling between the 3 models and can be changed in runtime according to the user needs by adjusting the weights of its neurons.

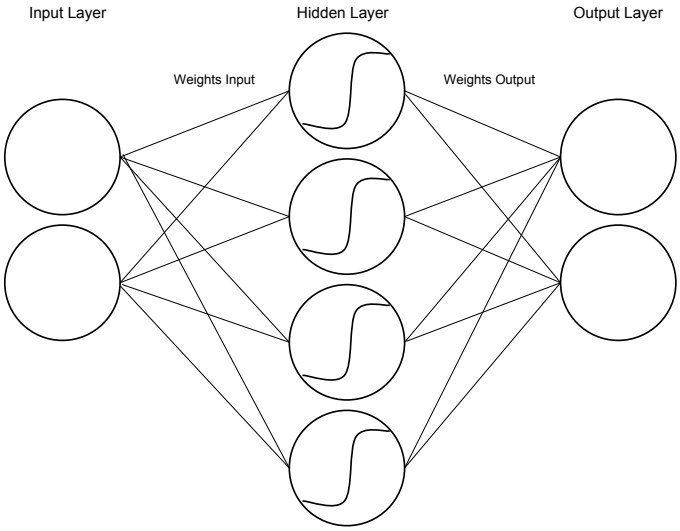



**Figure 1.17:** An Artificial Neural Network consists of artificial neurons. It has an input layer which takes parameters and a set of weights to bias before it gets evaluated by the hidden layer where a function, typically a sigmoid function, evaluates the bias input and outputs its state to the output neurons.

This process makes ANN less deterministic, but a lot more agile when dealing with inputs of all sorts and is a lot more independent when it comes to the environment they

are deployed in, meaning that they do not need the extensive supervision and design that AI implementations will. According to [Gaura & Newman, 2003, p. 101], ANN works well for capturing associations and discovering regularities in sets of patterns where the volume, diversity, and number of variables are high. However, it is harder to describe and verify why these connections are made. This does not discourage the authors from using ANN in a complex AHS; they argue that the Internet is made of the same kind of vague and scattered pattern as an ANN exhibits, and therefore fits the task a lot better. [Gaura & Newman, 2003, p. 103] implies that the most appropriate method for such a task is a Self Organizing Map (SOM), because SOM are reenforced learners that become more and more trained as they are exposed to an environment. SOMs work by assigning an optimal space within a radius to a given variable. The algorithm searches for the best fit within the radius and applies it. As the SOM is exposed to its environment, it reduces the radius where it can reassign its incoming value, and therefore it becomes more certain of the allocation of a given input. This works well in multiple trial environments, such as described in both [Gaura & Newman, 2003] [Papanikolaou *et al.*, 2000], where the educational context requires the system to find the users current experience and knowledge within a limited domain.

## 1.6 Part Conclusion - Analysis

The complexity of the data structure with the advents in information technologies is increasing concurrently with the increasing amount data, which requires new methods of organization and communication of the data. The most significant area of complex data is the Internet which holds a substantial part of the communication in the society and is therefore used as an example in this thesis. The rapidly increasing rate of data contributions and the enormous amount of data, requires a unifying method for representation of data in order convey meaning to the users, establishing a basis for knowledge distribution on the web. The ontologies describing the relationships between data are with the advent of Web 2.0, becoming an essential part of the Internet and is projected to grow even more important/essential for Web 3.0, which is the abbreviation for the Semantic Web. The amount of information that users can cope with is of the essence, because there is no need to show more than necessary to the user, as argued by Norman in his affordance dialectic.

This project focuses on the aspect of the threshold between a hidden and a perceived affordance of a given object in an interface - the term *meta-affordance* is introduced to accommodate for a relationship between virtual objects and can be categorized as "the value" of a search query. This term can be matched with a *User model* as the users' preferences or bias. This adaptive model for each individual user greatly reduces the perusal space to retrieve information/data from, but also limits the user in the information they retrieve from e.g. a query in a search engine. Therefore, detecting user behavior and ability would be of interest when constructing *User models* that are adaptable - determining whether an affordance in an interface is hidden or perceived to the user.

It is noted by [Gaura & Newman, 2003, p. 103] that agents based on Artificial Neural Networks (ANN) are good at traversing data and get hold of data - the problem lies within classifying the relations as "good" or "bad" which relates to the described *meta-affordances*. This problem is sought to be solved in this project by a generic calibration routine - consisting of a simple task to determine the user's optimal behavior to the specific affordance. Therefore, it is necessary to observe users in interacting in a given environment, with a simple generic task and determine if it is possible to make a general observation for optimal interaction behavior for all users. Such a description of optimal user behavior and whether disturbances / distractions can affect the observed behavior of the user - is of interest to this project.

## 1.7 Problem Statement

*Is it possible to detect differences in the user behavior, while performing simple tasks in an interface, changing only the complexity of the environment.*

# 2. Design

Detecting optimal behavior of the user is of great interest in order to expand a *User model* by using the principles from Adaptive Hypermedia Systems. The problem with the existing adaptation solutions is when using only predetermined preferences and user history, the risk is present of respectively over-complicating by making the *User model* with too many options or over-fitting the results by only showing information, which has been in the user's interest before. A solution to the problem would be to calibrate the *User model* to the user's current context, but former methods to generate *User models* have proven to be complicated and/or cumbersome. The general issue of generating *User models* are broad, but in this thesis, search queries performed in a data space with complex structures and relations, like the Semantic Web, will serve as foundation for further investigations. Therefore, this project will focus on a generic approach to calibration of *User models* as a client based solution - which can be re-calibrated according to the user's wishes. Specifically, the focus will be on determining whether or not user behavior can be detected in users' interaction with an interface in order to utilize this knowledge for a generic calibration routine of the behavior parameter of a *User model*.

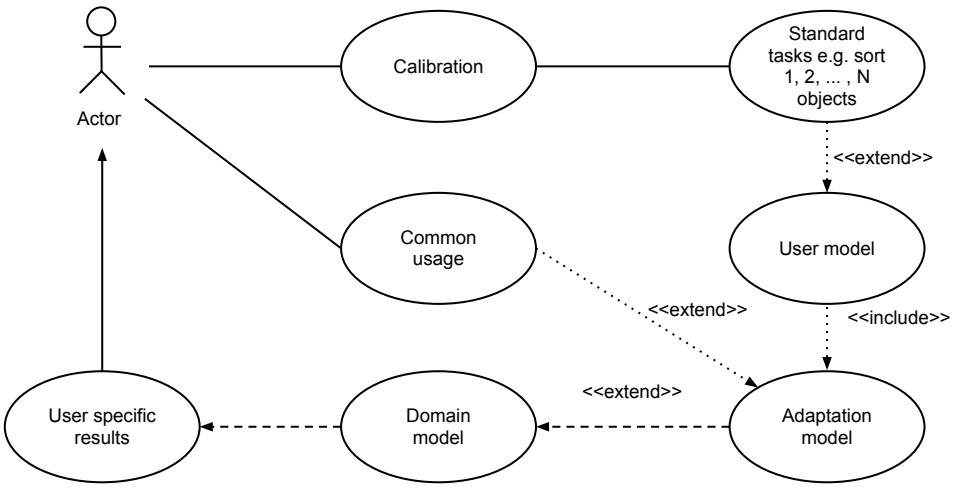



**Figure 2.1:** The proposed calibration routine can refine/define the *User model*, according to the users wishes by completing a simple task - determining optimal behavior of the user. This routine can detect inconsistencies in the modality between user and interface and adjust the results accordingly during common usage of the application.

Adaptive Hypermedia Systems (AHS) has proven to be successful on defined data sets, such as curriculums, since a well designed AHS will keep the challenges at a suitable level, according to the user's profile. These systems adapts the full set of data available to the application, to the *User model* so the user are only shown the amount of data, that they are capable of processing at their currently evaluated state in the system. If the principles from AHS can be applied to a dynamic data space, it can serve as a method

for fitting, e.g. the results of a search query to a certain *User model*. In order for this to work, it must be possible to distinguish the interaction patterns of users in a system in relation to their capabilities working with any kind of interface in any given environment. As proposed in relation to AHS, Artificial Neural Networks (ANN) have greater ability to adopt to undefined or infinite environments, compared to Artificial Intelligence (AI) solutions. This conclusion, in relation to the Internet based search example, would suggest that ANN would be an implementation method of choice. While an application utilizing an AHS implemented with an ANN, might be an effective and innovative method of a generic client-side user calibration, the approach is considered to be novel and requires thorough consideration of how the *Domain model* is fitted to the *User model* by using the *Adaption model*, as shown in Figure 1.12.

This project strives to find results that can be used for a generic and cost-effective calibration route so initial tests, to verify whether it is possible to determine differences in the interaction behavior pattern of users will be performed. The findings should be applicable for incorporation in other applications, such as programs and games, where the user behavior pattern would serve as calibration of a *User model*. However, for this thesis, testing users' interactions in specific programs and applications would be dependent on the experience (or lack thereof) of the individual user, leaving the results as dependent on the test subjects' prior knowledge of an interface or specific game, application, etc. To avoid this bias of test results, the investigations should be applied in a very simple environment, leaving no a priori knowledge as an advantage to certain user groups. Therefore, a new interface for the test must be devised where the tasks must be simple and trivial, exposing the test subjects to the same learning rate and difficulty level in the sequence of tasks. The only varying parameter being the environment where the user operates, such as additional affordances, different representations, and ordering of the layout of the application. This section will describe the requirements of the design of such an environment, and the tasks that the users are being exposed to during the test/calibration routine.

## 2.1 Implementation design

The modality between users and the affordances, hidden or perceived, of the interface must be determined through a calibration routine and is therefore highly dependent on design choices in the development process and the context of use. The design itself is going to be general and simple and in the context of this project, to prove that the method of determining user behavior is viable as a tool to other similar implementations which needs to calibrate their users' capabilities with an underlying more complex system. A browser-based solution will therefore be deployed in order to avoid compilation of the test application for a variety of platforms. This also ensures that the test results does not reflect the setting of the test which should be of the users' interaction with the interface and not with the test facilitators, the laboratory setting, or other factors that might compromise or bias the test results.

Designs of Adaptive Hypermedia Systems (AHS) is varied across implementation environments. The context is of great importance hence its specific standards across of these environments can vary, but the elementary design of an AHS is still fairly rigid as a concept, see Figure 2.14. This design proposal will be client-based so that it can be used with almost any web-service. It will need adaptation to be applicable in other similar systems but the underlying principles does not need to change as long as the purpose is the need to detect the user behavior.

The calibration routine also can be incorporated into the model because it will be operating in a stochastic non-defined environment, the *Domain module*, which is entirely server based. The remaining two objects are therefore the only modules that are calibrated and of interest in the design process. This is illustrated in Figure 2.14.

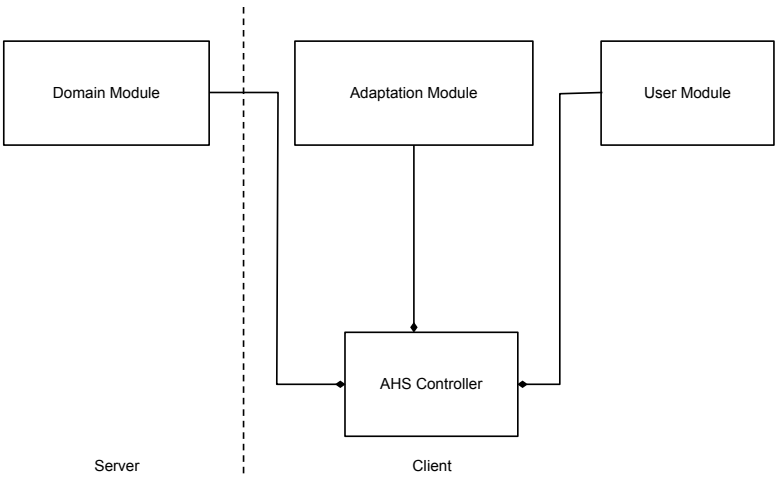



**Figure 2.2:** The adaptable test results adds an abstract evaluation layer, client side, to the application that can be updated during runtime of the application.

The calibration routine will be dependent on the design of the interface and its functionality. It will be deployed client-side only and the results will be delivered to a server prior to being altered by the adaptation layer. This is to ensure that the collected dataset is not being interpreted by such a layer that could bias the results - in functional applications this will be necessary to do client-side as well, to meet the requirements of an AHS.

### 2.1.1 Affordances in the interface

The affordances in the interface are small interactive bubbles that act as buttons which are the most simple interaction affordance. It can function as an entry parameter and does not need to communicate much in order to be understood as a button - it can be pressed, unpressed, visited, etc. It is important for the calibration routine to have a single simple affordance that it will be calibrated against, and not a myriad of different strategies that have different affordances embedded in them, because that might bias the test results as well. This might also cause trouble in completing the task posed by the calibration routine, and will have a different response type than the single affordance approach.

Adaptive Hypermedia defines the *Environment* as the *Domain*. The *Domain model* contains all the parameters that defines the environment - the more detail the domain model has, the wider a variety of controls defined in the *Domain Model*. The environment also encompasses any interface: Mouse, keyboard, buttons, touch interface, any given weather condition, social condition, etc. These should be interpreted in the client-side evaluation and reflected in the adaptation model as well but not be expressively represented either. Therefore, the meta-parameters that defines the *User Model* could be as follows:

- Interests | the personal interests of the user explicit or obtained from the user history.
- Connections | the personal connections of a given user obtained from the context.
- Behavior | the number of affordances that can be percieved by the user.
- Ability | the ability for the user to complete simple tasks.
- Etc.

The calibration routine only focuses on the number of affordances that are percievable by the user. Therefore, only such parameters will be of investigate, because the other information is already available from the webservices, and is not dependent on the current behavior of the user nor the interface currently in use. Because the environment of deployment is critical to the deployment of an AHS, there is a need to specify the environment of implementation, hence the initial problem is to devise and verify whether a simple calibration routine can be used to determine User behavior, as described in the Problem Statement 1.7. The nature of the task and the environment necessary to facilitate reliable

results to verify the problem statement also needs to be considered. The task must reflect the ability for the user to determine whether the affordance is perceivable or hidden to the user. The task is to sort a set of elements in the user interface, the varying parameter being the type of distraction that are designed to complicate the trivial task. The quantifiable observations from the test should reveal a difference between the types of distractions that users are subject to. The two parameters of interest when observing the users and the test participants will therefore be:

- Response time
- Completion rate

The set of tasks that the participants must accomplish are the same regardless of their affiliations or preferences which is done by developing uniform and simple tasks that are intended to all users, not necessarily the subgroup defined in this specific use case. This standard approach ensures that the method is applicable to other user groups. The task itself must have a meaningful relation to the method of visualization such that it has potential to reveal information about the users, behavior & abilities, and the limitations that the affordances of the interface present to the users. The test data of interest - or observations - has to be of a uniform, observable, and objective nature. These observations could be expanded in future work, but for the purpose of the test and verification of the problem statement, there is no need to expand the number of dependent variables. The objective method will be an observation of the time, the user spends between each input during the test, in the remainder of this section called, *response time*. Because it is a contextual test, the data must be represented as a set that a user understands, such as a set of numbers, the alphabet of their native language, etc. The second parameter is a correction parameter that reveals the users ability to complete the assigned task and the correctness of it. This parameter will be called *completion rate* in the remainder of this section. The completion rate will also account for the instances where users do not understand a task or cannot possibly complete the task, by only observing the user in a limited time frame, e.g. 20 seconds pr. task. This restriction ensures the continuity of the program and ensures that the test will be completed within a set time frame for all participants, even if there are problems with the interface or the visualization.

## 2.2 Task design

The design of the tasks should all involve a different layout on the screen where as the objects in the interface should be of a similar and uniform design. The objects should all be clickable by the user - both to communicate succesfull interaction to the user and to log the interaction time and completion rate of the user. The type of task can be any type of task depending on the interface that the test is calibrated to. We have chosen to deploy the test as a client-based application using a computer or tablet as a platform. A browser based solution beats a lot of implementation issues and there are well documented and established IDE's, API's to deploy such a solution within the timeframe of the project. The sequence of different visual layouts will be repeated with different distraction types to test whether there is a measurable difference of the user interaction behavior.

The task will be for the user to put enumerated circles in ascending order by clicking a set $S$ of objects in a 2 D interface. There will be a need of $N$ number of categories of visualization layouts containing the same task of sorting the visual objects in the interface in addition there will be 4 different distractions $D$ in the interface environment.

$$totalnumberoftasks = S * N * D \tag{2.1}$$

Keeping the same simple ensures that there is no understanding issues, such that the application measures the user's interaction with the interface and not the user's performance in a given test. The time that each user has for each task is 20 seconds, giving them an average of 2 seconds pr. interaction to respond, when $S$ is set at 10. This time limitation should both guarantee continuity of the test and ensure that in case of misunderstandings the test proceeds nevertheless.

### 2.2.1 Visual layout

The visual layout for all objects in each task should be distributed such that all objects on the screen has the same relative distance to each other according to dimensions of the screen space. This will make the tasks uniform to each user in regard to positioning on the screen. The objects on the screen can neither overlap or should avoid it by all means, the objects can furthermore benefit from being animated to get the users attention and making them less prone to leave the activity prior to completion of the test. The layouts are all 2 dimensional and should creatively be distributed such as to guide the user. Geometrical algorithms that can be generated recursively would be of great interest when designing a test for such a visual interface as the one needed to test user behavior. A few possible layout patterns are described in Figure 2.3, keeping in mind that the interactive objects must be clickable and visually distinguishable from the background.

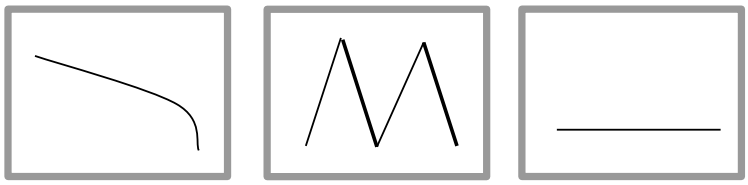


**Figure 2.3:** A few proposals for a simple layout strategy of a set of objects. The distribution can be set using a mathematical distribution such as the logarithmic & the linear distribution of interaction objects in the interface.

### 2.2.2 Distraction types

The distraction types are the indicators to whether there can be detected difference in the behavior of the test subjects given that the environment changes type, still facilitating successful interaction between interface and user. The 4 types of environments will be as proposed in List 2.2.2 below.

1. Without any disturbances
2. As an increasingly messy interface with additional objects for each interaction
3. Using size of objects as indicator of their numerical value instead of the character representation
4. A second order task where objects are introduced for each interaction - increasing in complexity

The reason to make these different distraction type of scenarios is to see whether there is a difference in response times. Therefore, it is important to emphasize that the user must have the ability to access the visual objects at any time. There is no "right" answers hence all objects are clickable at any time - it is up to the user to make the correct assessment of value of the objects, so no constraints should be on the interaction of the objects.

The interaction during the test should be logged with a primary focus on saving response times pr. task that means that a the logging system should be capable of saving all mouse location for an interaction, the correctness of the interaction, position, etc. The interaction should only be possible with a pointing device so there should be no need for additional handling of other types of events.

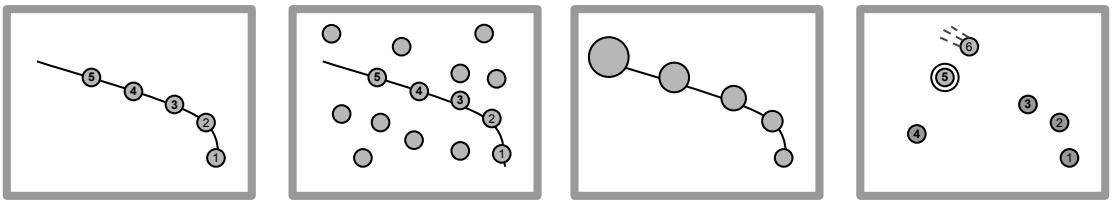



**Figure 2.4:** The 4 distraction types devised in the test: The upper left is without any disturbances, the upper right is a task where there is an increase in interaction objects without any relation to the task at hand, in the the lower left, the sorting will be conducting by size of the visual objects alone, where as the lower right is a 2. order task where the next object in the sequence are introduced as the user puts the sequence in order.

## 2.3 Test participants

When testing the hifi prototype, a relevant target group is to be decided among all possible participants in the test. Therefore, there is a need for defining the specific segment which is targeted, in traditionally testing like usability testing. Determining user behavior in general can be problematic hence people have different background, schooling abilities, etc. and therefore a more uniform target group is needed in order to verify if the method is suitable for detecting behavior of the user in the usage pattern of a given application. Determining whether the proposed approach of calibration of users' behavior and ability in a visual interface is a comprehensive task, is therefore done by choosing a subgroup with a well defined set of information *common factors*, that segments the participants by association of faculty, their current training, gender, age, etc. [Dumas & Redish, 1999, p. 120] states that when developing profiles of users for testing, there are 2 types of characteristics which should be taken into consideration:

- Common characteristics that all users share
- Specific characteristics that will vary among the users/user groups

For this project, a minimum of computer knowledge is among the required skills, since all students at Aalborg University work on computers for writing reports, solving mathematics, working with statistics, e-mail communication, registration for exams, etc. Also, a certain level of abstraction is expected since the admission requirements for attending the university is graduation from highschool along with the relevant subject specific courses. In the context of developing a specific product, an initial questionnaire for determining the profile of the possible participants in the test would be preferred, but for the initial investigation towards a generic method for individual calibration of an application, it is more important to look at the user patterns for detection of behavior inconsistencies, instead of explaining the underlying causes and reasons for a certain behavior.

The subgroup has to be available to the test team and must have general knowledge of computers and computer interaction in such a way, that the only novelty factor for the test participants, is the interface itself and not interfacing with a computer in general. The task and sequence of tasks will also be the same for all test participants, making the task and sequence a dependent variable in the test.

### 2.3.1 Academics - sub group

When trying to verify certain characteristics of potential user groups, one or more subgroup are composed of people who share specific characteristics that are important for selected profile [Dumas & Redish, 1999, p. 123]. The most obvious readily available source of test participants are university students at Aalborg University - where the authors currently

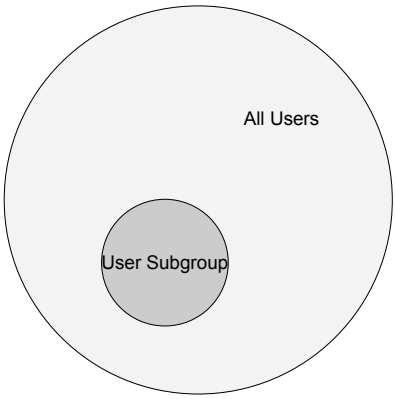



**Figure 2.5:** A test will determine if there is a detectable difference in behavior with the proposed approach for a subgroup of potential users. The demographics of the subgroup are students at Aalborg University whom all have the same schooling - apart from their current specialization, are in their 20'es etc.

attend. These participants are in general terms, in respect to the user group of all computer users, a subgroup that we in the terms of this project define as academics. The well defined subgroup can hence be defined to be of academics that currently is attending Aalborg University, all whom has a specific relation to both *faculty*, and *education*, the former being visualized in Figure 2.16.

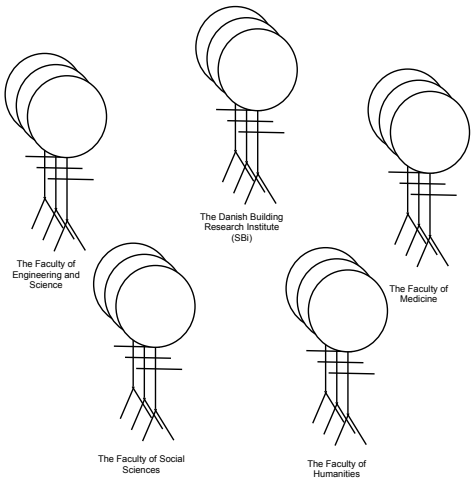



**Figure 2.6:** As a representation of the subgroup of academics at the University of Aalborg, AAU, there are 5 faculties that are all regular computer users and therefore a part of the larger user group in general.

These common factors are well defined, and depending on the current semester the students attend, they will have been exposed to not only the scientific method, which is

general to all attendees and affiliates at the university, but also to the specific approaches, paradigms and methods used and taught at the different educations. There is a risk in doing so - there can be common factors that are not associated by being academic - such as attending sports activities, playing computer games, prior studies at a different faculty than what the student is currently attending that reaches beyond the academic sub-group, which should be considered in the case of finding significant differences in the test-data. The segregation is visualized in Figure 2.17 below.

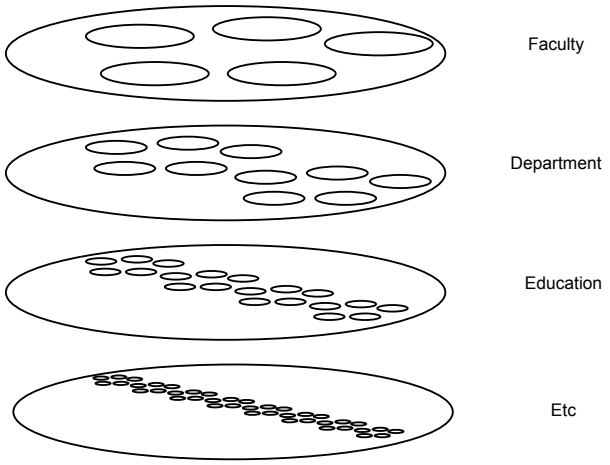



**Figure 2.7:** The common factors for the subgroups of academics are determined as a part of a questionnaire filled out by the participants as a part of the test.

This common, yet widely different, knowledge base are great *common factors* for testing an approach, hence different *schools of thought* has a very different take on the same topics. An engineer, an architect, and a social scientist can appreciate the same bridge construction and derive very different conclusions from it. The engineer will typically see a bridge as an engineering marvel, considering the strength of it in terms of physics, whereas the architect typically will describe it in terms of its design, and the social scientist will try and understand the implications of bridges on the societal structures.

Considering the broadest possible segregation between participants in the test, of the *common factors* - the students' affiliated faculty, there is expected to be a difference in response time and completion rate for the participants. Unless they have changed studies several times they will all have been exposed to 1 possibly 5 different approaches to methods of perception, quantification, etc. 5 will be considered extremely rare and not to be the

case, but 1 *school of thought* should be the norm for students participating in the test. The test participants will be asked the following information:

- Faculty
- Education
- Semester
- Gender
- Age
- Handedness

The remainder of common factors will be further engrained by their faculty affiliation, their specific education, their current semester, the gender, age and finally handedness which is a common trait for all users in general. This granularity makes it an interesting data set to acquire for determining whether a visual approach to contextual presentation is applicable to the general user group, and whether other applications could benefit from such an approach.

## 2.4 Test Assumptions, Constraints & Limitations

Since the metrics of user behavior & ability requires an actual interface to be tested, a working prototype must be deployed in order to verify, whether such an approach is useful or not for measuring and determining the interaction activity.

The description of the group of test participants, specified in Section 2.4, constrains the group to a small subset of all potential users. In addition, there are constraints, as described in the Appendix 2.2.2, because of the choice of test approach, which are web-based much like online questionnaires - the automation of test procedures are also a parameter in the procedure of determining a *User model*, since it can be scaled to larger user-groups and be used client-side as an aid for users, as well as it can be used as server-side to monitor user behavior in a given environment.

The overall motivation for the project is to investigate if there is basis for devising a method to calibrate an interface using very specific tasks dependent on the environment. This is done by measuring whether there is any clustering of the behavior patterns among the users. In order to constrain the problem to a tractable form, additional limitations and constraints are described in the next section, where the experimental setup will be described in detail to ensure reproducibility of the test for other researchers. The reasoning behind the assumptions, limitations and constraints of the experimental setup will be summed up in Section 2.5, where the exact test parameters will be described.

In order to constrain the problem field into a tractable, reproducible form, the assumptions, constraints and limitations of the test design will be described in this subsection to establish the foundation for a demographic test with a set of uniform, simple tasks.

### 2.4.1 Assumptions

The problem will be a visual indexation task where a logical distribution of numbered, clickable objects presented in an overall geometrical context as a visual cue, such as a spiral or tree. The type of tasks are indifferent but must be of a simple type to ensure that the type of tasks does not bias one segment of the test results due to their specific knowledge base. The following assumptions, listed in List 2.5.1 below, are tested:

Asynchronous laboratory experiments can be used and the experiments shown in Appendix A shows that critical errors are caught as efficiently as in a laboratory experiment or other manhour intensive procedures. The assumption of general group vs. subgroup testing is that the demographic are all between 20-30 years of age compared to the population of users as a whole. They have all accomplished the same basic requirements in order to gain access to the university. They have different social backgrounds, capabilities, preferences, interests, etc. but they do, however, all use computers on a regular basis. The parameters of interest in this study will be divided into two different types of information the information about the user's and information about how well they performed the tests.

The user information will be anonymous and primarily concerned about data relating to their performance, demographic relationship with the faculties at AAU, their age, gender, and other generic data that could influence the results. .

- That a *User model* can be generated from a simple calibration routine.
- Remote testing can find critical errors in interfaces - but can it find the subtle differences in the behavior of a user? The initial tests will be carried out with facilitators to ensure that the test subjects complete the test.
- The demographic is fairly narrow and has a similar age, yet they have the same basic training but different knowledge base, and a lot of other demographic differences where we only cover basic information. This difference will be larger in the general user population.
- The interface is a visual indexation tasks with a pointing interaction device as utility.
- Task indifference - but should be as simple as possible.

For the test, it will be assumed that all participants have no disabilities in common and if they have, it will be considered a random factor along with the other constraints. The participants will be randomly distributed and if enough test participants are involved, these disabilities should have a random distribution across the population.

### 2.4.2 Constraints

The constraints in the current setting will first be derived from the generic factors that can influence the results and will be considered as one dependent variable, while in reality it could prove to be quite an assumption to think of all generic factors as dependent - the test can be designed and devised in such way that it is equal to all test subjects. The generic factors in the experiment are:

- Screen size (width, height, resolution)
- External factors (lighting, interface, hardware, responsiveness)
- Internal factors (habits, disabilities or cognitive limitations)
- Learning (prior training, education, interests - than that of the test subjects current affiliation with faculty & education. )

These black box variables can be considered if there is a detectable difference in the test subjects response but until this difference is detected it will be considered as dependent variables. The explicit information that is to be obtained from the test subjects is going

to be demographic information that verifies that they all are in the same user group and differences in their responses are not due to age difference, e.g. due to lowered sensibilities due to age. This simple anonymous information about each subject is collected so that it can be made public.

1. Age {20,...,30} years of age
2. Gender {F, M }
3. Study {Open text-field, there are too many studies to list}
4. Faculty {hum, samf, tek-nat, sundhed, byg }
5. Current Semester {1,2,3,...,11}
6. Handedness {L, R}
7. Device {mouse, trackPad, touchScreen, trackBall }

The information obtained from the tasks that the users complete will all be quantifiable in order to make the *User model* applicable to the calibration routine and capable to be adjusted to each client and/or user. These features, shown in List 2.5.2 below, can be extended but for the purpose of this specific project where performance in the test interface is of interest, the features will be the following:

1. Task response time
2. Error rate - difference in order of answers compared to the proposed task

The assumptions about the interface will be further constrained by using a traditional computer setup for the prototype test will cause a number of constraints to the user, e.g. the Gestalt principles that influences perceptual interpretation of the visual representation of the test subjects and therefore the ability to answer results. As long as the design of the interface and tasks in the test are all the same for the population of test subjects such that the treatment parameter can be considered not to be weighted in favor of a specific segment:

- Relative screen position of the object (occlusion, movement, size) 2D representation removes most of these issues.
- The task objective - a uniform objective in all tasks sort from smallest to largest
- Visualization method - is of no concern as long as they are equal to all test subjects.
- Object representation ( size, enumeration, color )

The interface is a web-application meaning that it functions from a browser and therefore does not have to be compiled for the different test platforms and devices like laptops, tablets, pads, etc. and has access to the web and other features required by remote testing of users. This method of testing can be done with graphical interfaces in general but the method should in theory also be applicable to other kinds of interfaces and representations.

Limiting the variability of the interface and task procedures should limit the influence that these factors have on the test results - the choice of visualization strategy will also influence the results and a minimalistic and simple approach will be the primary constraint on the interface for the purpose of testing but could be very different in other applications.

### 2.4.3 Limitations

The limitations of the experiment should be concerned about what the test does but cannot account for such as the environmental constraints that influences all interactions between a user and an interface. Hence the prototype will function in the same environment, the browser, as potential applications, it can be treated as a dependent variable as a whole and will be considered equal for all subjects. This can, however, limit the conclusions, e.g. if the test is taken in sunlight that reflects in the screen making the visual representation impossible to see.

If there is a standard task and all objects have the same affordances, the task itself is the independent variable in this experiment. When measuring the user's ability to solve a given standard task, the parameters to determine the user ability are time and error rates as mentioned in the constraints, 2.5.2. The representation will be in 2D because there are fewer freedoms that could influence the results and there are formalized parametric estimations of user response times depending on complexity of a task in such an environment. The overall limitations of the tasks that the test subjects must solve are:

- 2D representation
- 10 interactions pr task
- 20 seconds maximum completion time for each task
- Avoid colorization of objects/or keep to a minimum in order not to bias people with color sensitivities/insensitivities.

The items in the list 2.5.3 above, are all specific for the tasks at hand and could possible be too difficult to complete within the set time-frame. Since it is equal to all users, there should be no bias toward any specific demographic segment unless they have a tendency to solve a problem by another strategy, more time-consuming strategy than others, which then should show up in the evaluation of the test results. There will also be an option for the users to quit the test if desired so. This will be a hidden affordance to the users but

is a requirement to the interface in order to detect the users that do not want to complete the tasks. Hence it can be a parameter of segmentation - how many users who have not completed the full test.

## 2.5 Part Conclusion - Design

The method that is proposed in this project is a generic and cost-effective calibration routine to describing user behavior & ability without the dependency on history or complete representation of the environment. Determining the behavior in the users' interaction with a given system, by calibration using a simple task, eliminates the need for a complex set of observations about the users and their environment, by observing the user patterns and thereby calibrating the *User model*. The test is devised to determine whether the user is capable of utilizing the potential *meta-affordances* or objects in the interface.

Initial tests are needed to verify whether it is possible to determine differences in observations of users' behavior in a closed environment and if any interaction patterns can be determined. If that is the case, detection of behavior should be applicable to other demographics and scenarios than the one proposed and described in this section. Laboratory testing would be very time consuming and costly, and research have shown that remote testing is performing very well. Furthermore, a evaluation of user behavior of an application, could risk being biased by an unnatural setup in a laboratory and testing by visiting the different users at their offices could very well be felt obtrusive, stressful, demanding, and inevitably very time consuming for the test facilitators and participants.

For this project, a low fidelity prototype will be insufficient because measurements in an exact environment is needed to avoid analogue differences, and a high fidelity prototype is therefore developed as a test suite, where the relevant observations can be made. Testing in specific programs and applications could possibly bias users who know these systems in advance. The high fidelity prototype will therefore be made very simple and non-biasing in terms of subject specific task, which certain subsegments could benefit from.

The test itself is designed to reveal differences in user behavior - more specifically the response time between interactions and their ability to sort a visually dispersed set of objects. There are distractions introduced in the interface to compromise "optimal" behavior so that the effects of these distractions can be observed in later data processing.

# 3. Development & Implementation

This section will describe the implementation of the high fidelity prototype outlined in the Design Chapter 2. The design suggest that a web-based high fidelity prototype for testing is implemented in a similar environment as where possible applications will be eventually deployed. These requirements means that the implementation should be a high level language such as Java, JavaScript, Ruby on Rails, or C++. Cross platform development should also be considered hence multiple compilations of the same program to different platforms could introduce unwanted implementation issues for the high fidelity prototype. Therefore, the technologies used are all multi platform compatible so all the platforms could be deployment/development environment.

## 3.1 Implementation of the high fidelity prototype

The choice of implementation platform for the prototype has to be in an easy accessible cross platform environment to facilitate the requirement of remote testing of the interface and being able to perform detection of client-side user interaction behavior. The choice relied on a flexible and integrated that could operate on several platforms and therefore an internet browser web-application environment was chosen. This choice left out Java and C++, and left the team with a choice between a variety of web development environments such as JavaScript and Ruby on Rails. Because JavaScript is supported in most modern browsers along with the advent of the HTML5 standard supported by most common browsers like Mozilla Firefox, Apple's Safari & Google Chrome, there are rich possibilities in this environment to make multi-media web-applications - which fitted the development team's skills and prior experience well. The numerous APIs developed to the HTML5/JavaScript canvas has been in a riveting development even though the standards are still in development [W3C, 2011]. The development process has involved looking at several APIs that are based on the HTML5 canvas are listed below in Table 3.1:

| API | 2D | 3D | DOM-support | svg-support | source |
|---|---|---|---|---|---|
| three.js | - | + | + | + | [mrdoob, 2011] |
| processing.js | + | - | - | + | [Resig *et al.*, 2011] |
| raphael.js | + | - | + | + | [Baranovskiy, 2011] |
| paper.js | + | - | + | - | [Lehni & Puckey, 2011] |

**Table 3.1:** Visualization APIs using HTML5 canvas

In order to facilitate the different visualization APIs that each are in a continuos developing process, some still not out of their beta-stage, an implementation solution was chosen where the rendering of the affordances are decoupled from the system implementation. Hence a 2D representation is described as the preferred solution in Section 2.5, it leaves out the three.js API - and makes the choice of API a question of taste and skills according to the wishes of the developers. The choice being processing.js because processing

comes with a more extensive set of documentation & examples than the other 2 available options.

### 3.1.1 Test environment

The test environment is defined in Figure 3.1, and is composed of a particle system which will be described later in this subsection. Each particle contains a set of attributes that defines their position, velocity and acceleration, color, radius, and id. Finally, each particle has an information field where hyperlinks, text, image-references, etc. can be stored. The behavior algorithms will control the particles' behavior within the environment and will be evaluated in run-time to avoid that particles with different affordances block each other and acts as hidden affordances. The strategy algorithms are a set of recursive algorithms that implements mathematical functions as a display strategy for the layout for each task, described in Subsection 2.2.1. As mentioned in earlier sections the visualization method is not of any particular importance as long as it is the same for all user-groups. Finally, the System Controller will have a logger module for tracking the user interactions and a module for controlling the sequence, the tasks and layout strategy during the test sequences. The logging mechanism and approach is described in Section 3.2.

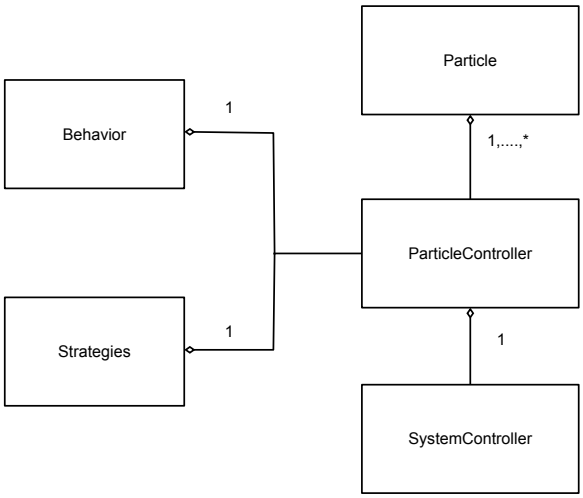



**Figure 3.1:** Each particle is controlled by the particle controller which has a behavior algorithm and a visual strategy class to set different visualization strategies for the particles. The system controller controls the test parameters and logs user input.

The environment that the test subjects will face will be a particle system implementation, where the objects will be reactive to their environment and each other. This approach is defined in [Parker, 2007], where changes in the velocity vector for each particle in the

environment will be defined by a set of rules as follows.

$$velocity = \sum_{i=1}^{n} rule_n \tag{3.1}$$

This algorithm will give all objects in the environment a behavior that can be manipulated at the request of users or developers. The velocity parameter is simply evaluated for each particle by the rules that are defined by [Parker, 2007]:

1. Boids try to fly towards the centre of mass of neighboring boids
2. Boids try to keep a small distance away from other objects (including other boids)
3. Boids try to match velocity with near boids
4. Limiting the speed of each boid avoiding graphic artifacts & other affordance issues
5. Bounding the position of each boid to stay within a confined area.

All of these rules have specific attributes and we will not go into them all but an example will be explained to illustrate the inner workings of the boid algorithm. The set of rules can be expanded further to introduce predators, or perching where the boids achieve a steady state where there are no changes in their velocity etc. As an example lets look at the initial rule *Boids try to fly towards the centre of mass of neighboring boids* in 3.1.1. The rule implies that the center-of-mass is calculated and fed as a parameter to the function that guide the boids toward the center of mass (COM) of all the boids in the system.

$$v_{COM} = \sum_{i=1}^{N} boids.position_i/N \tag{3.2}$$

The resulting vector $v_{COM}$ defines the position were all the boids will flock to. For each individual boid it means the rule must calculate the vector to the $v_{COM}$ and return that resulting vector. The tendency to flock can additionally be controlled by a weight parameter so that the equation for $rule_1$ will look as follows:

$$boid_{rule1} = (v_{COM} - boid.position) * weight_{rule1} \tag{3.3}$$

The weight parameter is implemented so that each rule has a weight in the range $\{0.0, ...., 1.0\}$ which gives additional expressive control of the behavior of the objects in the interface. This control can be forwarded by the developer to the user as an option of control to the system. This will imply the following changes to the initial Equation 3.1

$$velocity = \sum_{i=1}^{n} rule_n * weight_n \tag{3.4}$$

It can additionally be noted that rules that relies on data-relationships could be interesting to implement hence they will have a direct influence on the behavior of the individual boids, say if there are relationships between the content of two objects this could be visually expressed in their flocking behavior. This way the mentioned meta-affordances would be interpretable to the user without expressively showing it in the interface as a text-based parameter.

The individual boids can also be considered agents that can have objectives as proposed by [David & de Castro, 2009] where the boids are used for cluster analysis in a data-mining process, this is quite an interesting angle on agent-design, where an affordance in an interface can have behavior and a set of goals at the same time. This way a user could have $n$ number of agents working for them as they use a given interface, as for the initial iteration goals of evaluating demographic relationships through a simple calibration routine it will have no initial value but should be considered in later iterations as an aid to the user.

### 3.1.2 Test tasks - calibration tasks

The test types should be uniform and be presented in the same sequence for all test subjects as stated in the assumptions in Section2.5. The tasks types based on the visual layout, will be embedded in 3 different environments. The 4 environment, will not have a visual layout but particles placed at random and a behavior that introduces a new object to the environment for each successful interaction.The visual layout are primarily based on the implementation that distributes the objects in the environment which is described in subsection3.3. In order for making the strategies task work the boids algorithm gets an additional rule added that makes the boid move toward a position that gets defined by the strategy algorithm:

$$v_{target} = (position_{target} - boid_{position}) * weight \tag{3.5}$$

The strategy algorithm is invoked every 20 seconds as described in 2.5 or when the user has clicked all objects which must be clicked. There are 8 strategies pr task which are:

- Strategy task
- Messy task
- Size Task
- 2. Order Task

The initial strategy task only involves 10 affordances that must be clicked in an ascending sequence, each object in the interface are labelled with a number to infer its relation to the other objects in the interface. The messy task involves envoking extra objects, without

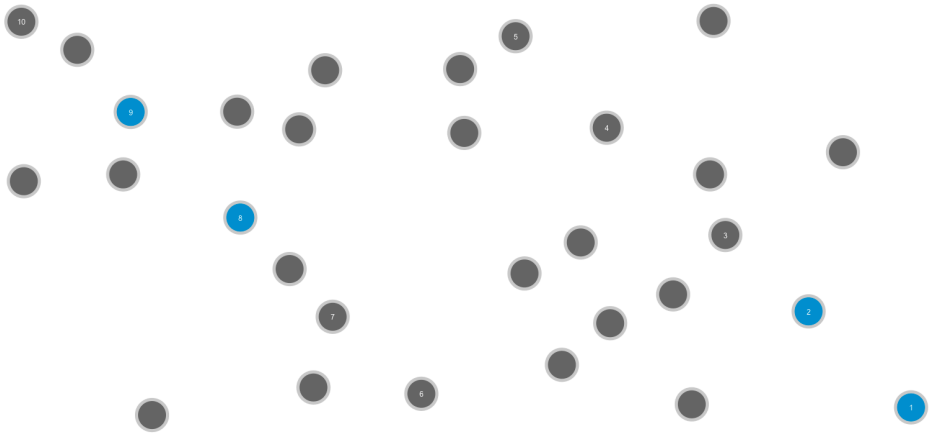



**Figure 3.2:** A screen dump of the MessyTask, where additional objects / affordances are introduced each time the user interacts with an object in the interface. The task goal is to put the labelled objects in ascending order.

a label, to the interface each time the user clicks an affordance in the interface, the boid rules ensures that the newly instantiated objects do not occlude each other but still clutters the interface with hidden affordance objects.

The SizeTask does not clutter the interface with additional objects but it removes the label on each task and uses the radius of each object instead as a visual indicator of its relationship to the other objects in the screen. Thus that the smallest object interface relates to the 1. element in the sequence which should be interacted with by the user. The 2OrderTask differs from the initial 3 by not invoking the visual layout instead it is a visual complexity task with increasing difficulty pr. task. The goal of the task is still to put the objects in the interface in ascending order according to their label, which is reintroduced in this set of tasks. As the user pushes the object another object is instantiated into the environment at a random location. For each subtask an extra element is introduced in the beginning to increase the complexity of the task. This task is rather unnerving and requires the full attention of the user.

The tests are designed to both be challenging and simple, in order to get a greater insight into the user's behavior & ability. The system tracks the time that the user spends on each task, the errors they make and the overall progress in tasks. The user will have the ability to shut down the browser window and end the test at any time - this will be tracked as well and the results the user have achieved will be saved to the database. The process of acquiring data and the structure of the dataset will be described in the next Section 3.2.

## 3.2 Data Collection

The purpose of conducting the high fidelity prototype test is to collect data, which can be used to verify the problem statement, and determine if behavior and ability can be detected in user interaction. The full interaction pattern of the user is therefore of interest, since abrupt interaction behavior like breaks and pauses are very distinguishing for use patterns. The implementation of a logger, which records all user behavior along with information about the user and the environment, is therefore a necessity to the prototype. The logging of data is performed client-side in an asynchronous interface, which ensures that the user's network connection is not an issue, since no data is transmitted between the client and the server during the test session; though, the client system needs to process the set of tasks in the preferred browser application and therefore system lag will occur. In order to measure the correct user behavior, system lag is therefore recorded and can subsequently be subtracted from the interaction time or be used as a parameter for sample validity.The users must be recognized in relation to the interaction behavior in the prototype system to compare the results with the user demographics. In order to analyze the potential activity clusters and labeling to certain demographic features of the user segment. The user's are prompted by a disclaimer explaining the nature of the experiment and they give permission to share the data they submit anonymously. The user will then be asked to describe themselves under the following parameters which is then stored, as mentioned in the requirements in Subsection 2.5.2:

1. Age {20,...,30} years of age
2. Gender {F, M }
3. Study {Open text-field, there are too many studies to list}
4. Faculty {hum, samf, tek-nat, sundhed, byg }
5. Current Semester {1,2,3,...,11}
6. Handedness {L, R}
7. Connection device {mouse, touchpad, touchscreen, trackball }

Beside the demographical information about the user, information about the environment is also necessary in order to clarify how the measured results are obtained. This information is based on information about the client - software and hardware conditions. The data is available from the browser vendors, using Javascript, and can be accessed without involving the user.

- Screen resolution

- Operating system
- Browser application/version

The information about the user and the client serves as a description of the interface the user had available at the time of taking the test. The interaction datalog structure, looks as described in Figure **??**

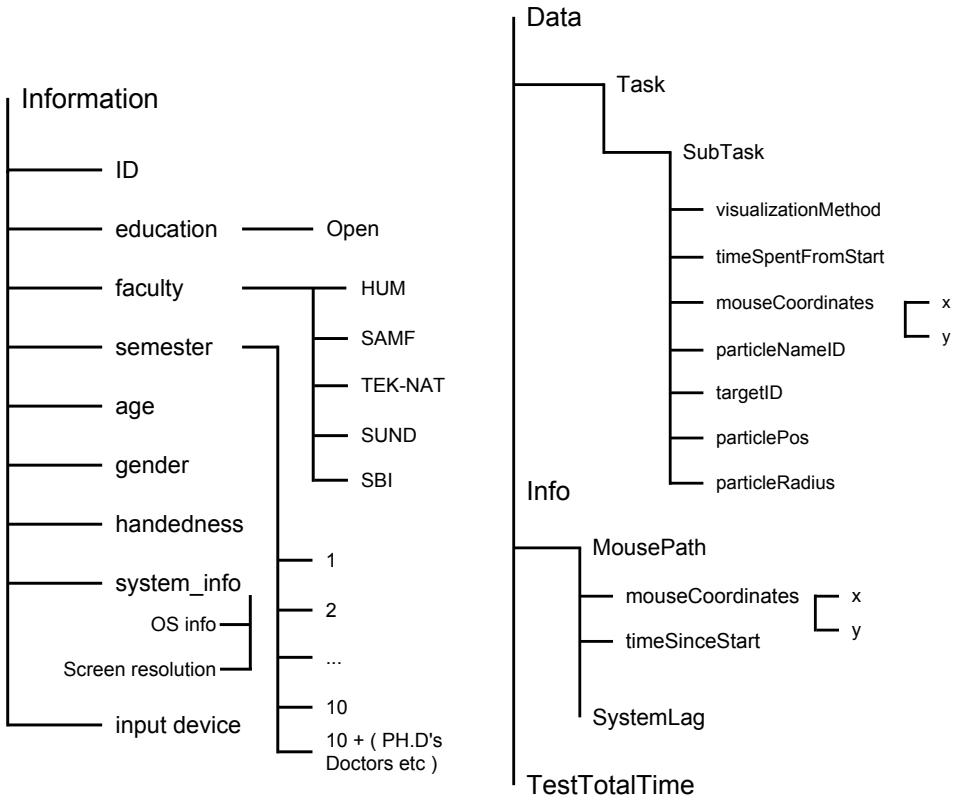



**Figure 3.3:** Right : The data submitted by each user from the questionnaire that all participants have to fill out prior to taking the test. The fields without description such as age, gender, handedness are implicit and the system_info is pulled directly from the browser without querying the user.Left: The data collected from each user, during the interaction with the interface. Each type of task will have a separate entry in an array structure, so will the Info data field. This data will be stored using JSON - strings in a the results field in MySQL database

The actual test is based on a set of simple tasks, as described in Subsection **??**, which will gradually ascent in difficulty as the test session progresses. The tasks are abstract without any subject specific knowledge required in order to avoid any bias; the sequence of tasks are equal to all users, so the results are comparable. The tasks are varying in types and complexity so the interaction with the interactive object is measured and stored per subtask as shown in Figure 3.3. In order to analyze the individual tasks, some information

must be stored for each task. First of all, it is necessary to identify how the task was shown to the user. In order to set a time limit for solving the task, there must be a timer from the start of the task.Two parameters must be logged such that the error parameter can be calculated. Because the performance of the system might be different according to the specific hardware, the lag of the system must be stored, so it can be subtracted for comparison between tasks. Finally, the total time spent on the test must be known in order to compare the test execution of the users. Therefore, for each task the following information is stored:

- The visualization method
- Time spent from start
- Mouse coordinates
- Particle name ID
- Particle pos
- Particle radius
- Mouse path
- System lag
- Test total time

The usage pattern of a single user can vary on a large scale; some users might complete the whole set of tasks either within the time limit or by timeout, while others might quit during the process, due to several reasons. The different behaviors are important parts of the use pattern and should therefore be logged.

The Figure 3.4 above, depicts the possible end conditions of the system - the user can either complete the full set of tasks within the maximum set time frame, complete some of the tasks but reaching the maximum set time frame before completing the task, or aborting the test session before reaching the end of the full set of tasks. These different user behaviors are essential for detecting user behavior, which is the purpose of the test. Therefore, the prototype must address all these end conditions, while still saving the data. Since the test is performed client-side, the data is continuously stored in the HTTP-session, which is an open stateful token-based connection between the client and the server, temporarily storing the data as a dynamic variable on the client. Using an asynchronous approach, no interconnectivity between client and server are necessary during the test session as described in Figure 3.4. When one of the end conditions are met, a function for submitting the data stored in the HTTP-session to the database, are called saving the collected session-data to the server side database. By saving the data about the user behavior during the

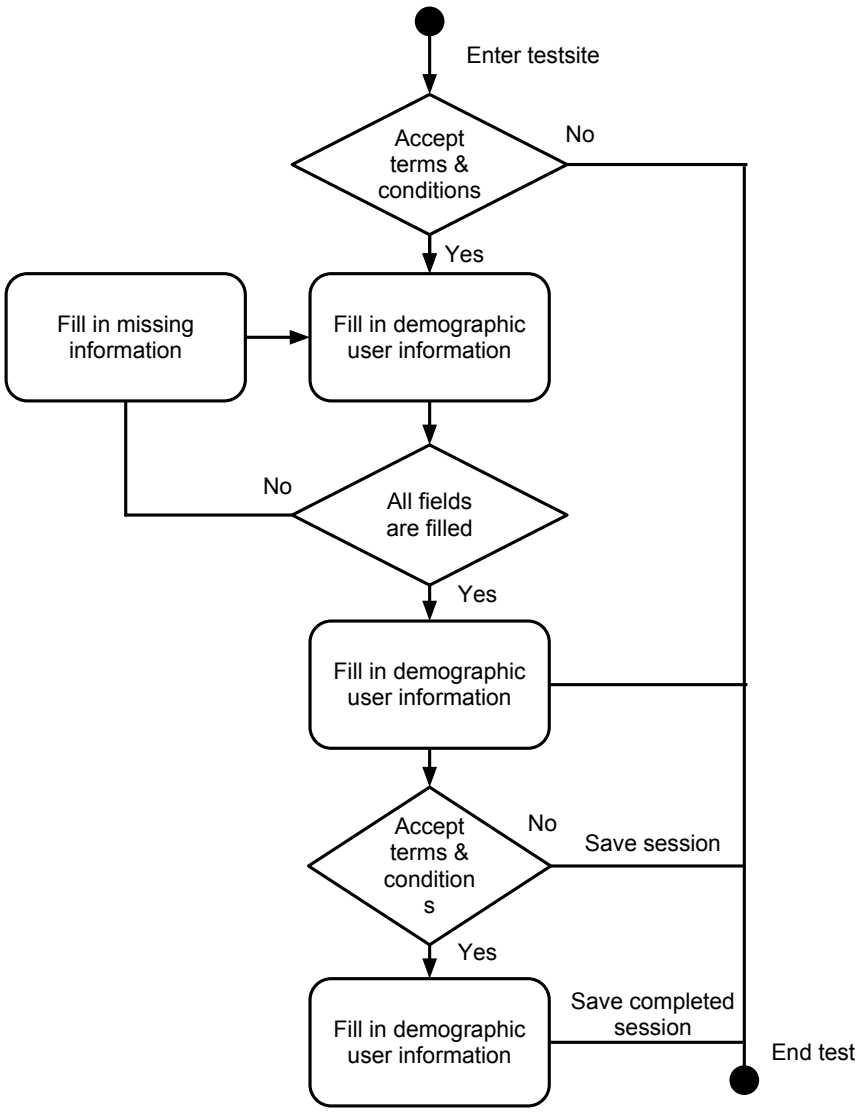



**Figure 3.4:** Flow chart of the prototype for the test. The user must accept the terms and conditions due to the logging. Demographic information is entered and form field validation secures that every field is filled out - alternatively, the user is looped back to the form. When the demographics are filled out, a HTTP-session is initiated where the data is stored and the user is sent to the start page of the test routine. The test is initiated by clicking the screen and when one of the end conditions is met, the HTTP-session is stored in the database.

test, it is ensured that all important information is saved regardless of how the user ends the test - by completion or abortion. The only way data can be lost is if the connection between the server and the client is permanently closed before one of the end conditions are met in which case the database will never receive an entry and the test data will be lost.

The implemented prototype for the test is described by a flowchart in Figure 3.4. When the user enters the site, terms and conditions accepted due to the fact that comprehensive of user data is performed. After filling out the demographic user information, the user is either looped back if that information is missing, or sent to the start screen of the test, where the tasks are initiated. When the demographic information is entered, a HTTP-session initiated and the information is stored within it ready for submission when the user meets the end conditions described earlier. When making a logging application, [Dumas & Redish, 1999, p. 191-192] list a number of issues to address:

- Each event must have a preset code
- Each event must have a time-stamp
- The duration of each events must be computed by measuring start and stop
- Keep track of multiple and/or concurrent events
- Data should be easily stored and backed up
- Data should be exportable to statistics applications
- Time should include various formats

The tasks in the prototype are all labelled, as shown in 3.2, and other events are mouse clicks and mouse movements, which are also labelled. Every event in the prototype are time-stamped, including start and stop of prolonged events. Because everything is stored temporarily in the session, multiple events are recorded on the fly before they are stored in the database when reaching one of the end conditions, which also addresses the archiving and backup issue. The data is stored in the JSON-format, which is compatible with a wide range of applications, including the selected statistics application. Time is recorded in miliseconds from the start of each type of task, so each type in the sequence has its own timeline. Client side statistical processing could be a possible use of eventual findings in this project. Since the datalog structure format is exportable to statistics applications through JSON - the files are stored without further processing & the calculations are easier processed in these programs than in the native prototype.

## 3.3 Task layout & description

The there is a total of 8 task layouts in the prototype devised for the test. They all distribute the objects on the screen using two processes a distribution process and a scaling process so that the distribution follows the same pattern regardless of the browser window size. The scaling is primarily in place to ensure that all test participants are subject to the same treatment for the purpose of the test.The distribution goes between highest and the lowest value in each dimension of the layout distribution, divided by the window proportions which are defined between $(0, window_x)$, $(0, window_y)$ which means that the most extreme position of the browser window are defined by $(window_x, window_y)$.

There are 10 objects in each scene in 3 of the 4 categories of distractions, the 4. task requires random positioning of introduced objects. The 4. category has 8 tasks, only differing by the number of known objects in the scene at initiation, which are placed randomly on the screen as the user completes the tasks. That totals the number of tasks that all users can complete according to the equation defined in 2.1, with a total of total number of interactions at 320. The visual distribution of the tasks are governed by simple algorithms that gives each visual object an x & a y-value. Where x is the input variable and y is the output variable

$$coordinate_{x,y} = (x, f(x)) \tag{3.6}$$

The visualization algorithms used in this have been selected from their aesthetic by the development team, hence no strict description was required in the design chapter **??**. The test visualization representations are as follows:

| | |
|---|---|
| 1.*arc* | 2.*sideways* |
| 3.*logarithmic* | 4.*exponential* |
| 5.*binarytree* | 6.*spiral.* |
| 7.*zigzag* | 8.*linear* |

The visualizations such as the logarithmic & exponential distributions are self-explanatory, hence x is fed to the built in Math function in javascript - which generates the y value of the coordinates for each object. The linear distribution is distributed along the x-axis, keeping the y - parameter constant for all visual objects. The zig-zag distribution is similar but offsets every even number of object on the y - parameter, using the % modulo operation. All the visual layouts are implemented recursively, for code aesthetics and the challenge of writing recursive code, the specifics of the layout is however not particularly important in regard to the performance of the user, apart from obvious obstructions to interactions. These obstacles are sought avoided in the behavior of the elements in the interface described earlier in this chapter in the description of the test environment 3.1.1.

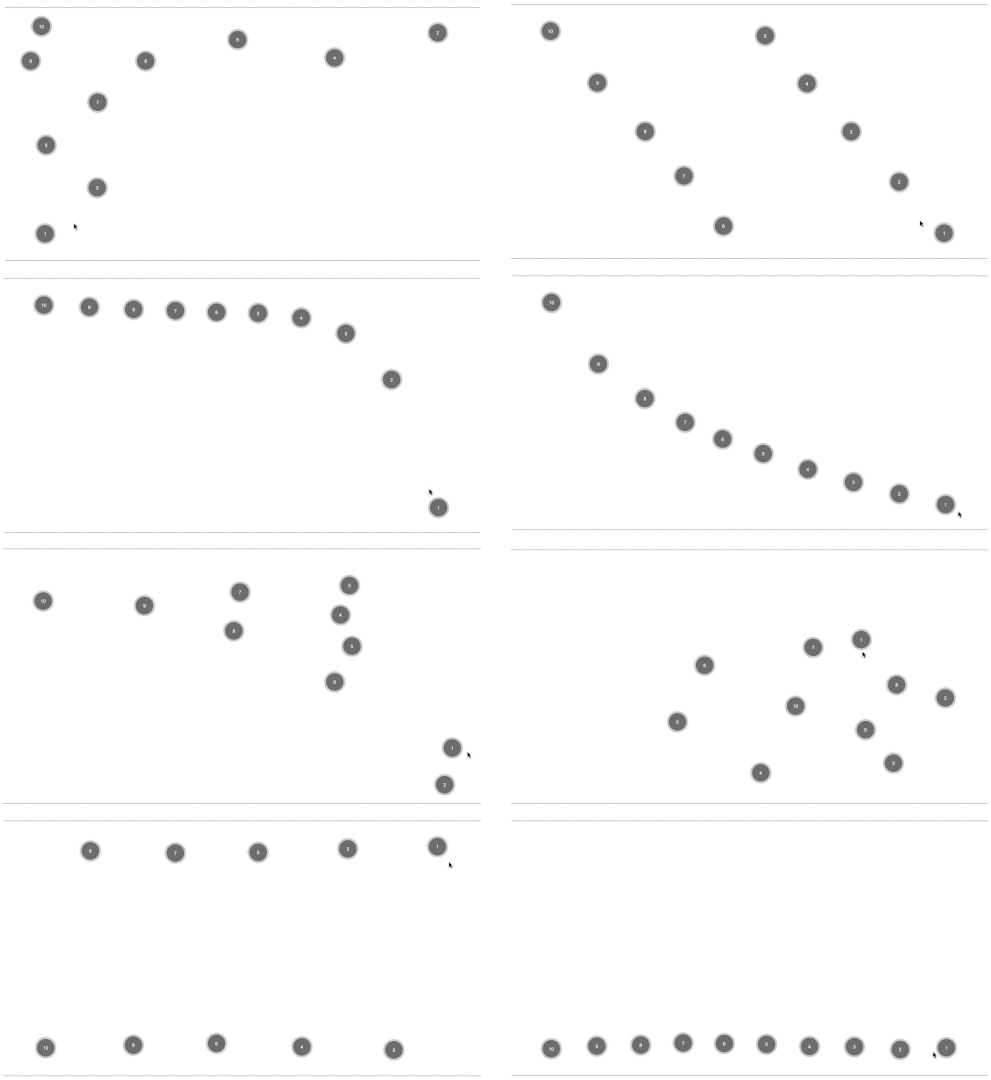


**Figure 3.5:** A screenshot of each of the 8 task layouts, the layout positions are computed and scaled according onto the screen, to ensure a uniform layout for all users regardless their platform. From the upper left to lower right the visualizations are : arc, sideways, spiral, exponential, logarithmic, binary tree, spiral, zigzag, linear.

## 3.4 Part Conclusion - Implementation

The implementation of the high fidelity prototype is a an effort to ensure consistency in the results to verify whether a simple calibration routine can be used to detect user behavior by quantifiable results. This have required the development to be focused on an embedded browser based web-application solution that facilitates the following:

- A setting that is as close to regular use as possible
- The ability for the users to quit the test either by closing the test, or by timeout.
- That all test subjects are exposed to a new and neutral interface, reducing bias and giving them an equal learning curve.

The system itself is a 2D representation of objects, composed as a particle system that enforces that the objects stay within the boundaries of the screen and do not occlude each other. The boids algorithm give the particles behavior and the different subtasks in the high fidelity prototype have different layout strategies according to a recursive use of simple math function, which are scaled to the screen resolution. The tasks themselves are of 4 different types but all have the same goal - to sort a set of objects in ascending order.

In order to log the user's interaction with the objects, response times for each interaction is logged as is the error rate, in the form of a targetID, as is the over all time spent on the test. The additional information that is being logged is the system lag i.e. the refresh rate of the canvas, the resolution of the screen and the operating system information. The user is prior to testing being required to accept the terms and conditions for the test which includes publication of the data-set, all information will be anonymous. After accepting the terms and conditions, the user is prompted to fill out a form that describes their faculty association, their current semester, their education, their age, gender and handedness which must be filled out prior to initializing the test. Each type of task in the test takes 20 seconds so the user will have approximately 10.5 minutes for the whole set of tasks. This might not be the case for all users so they will have the non explicit option of leaving the test at any time and still make an entry into the dataset. The reason for doing this is to see if such a behavior also could be deemed an interaction behavior pattern.

An interesting finding in the implementation stage is that the boids algorithm have been used in data-mining clustering analysis in the work of [David & de Castro, 2009], where the boids have an AI that collects data that adjust the weights of the different boid rules. This approach of intelligent agents as an asset for acquiring data for the users, seems very promising and could prove to be interesting to deploy in later iterations.

# 4. Evaluation

The approach for verification of the Problem Statement 1.7 will be through testing of the implementation described in the previous Chapter 3. The test was conducted online and distributed by email to all the faculties at Aalborg University in the period from the 28th of September to the 5th of October 2011. A total of 300 participants were detected as entries in the database, see the distribution in the test result Section 4.1.

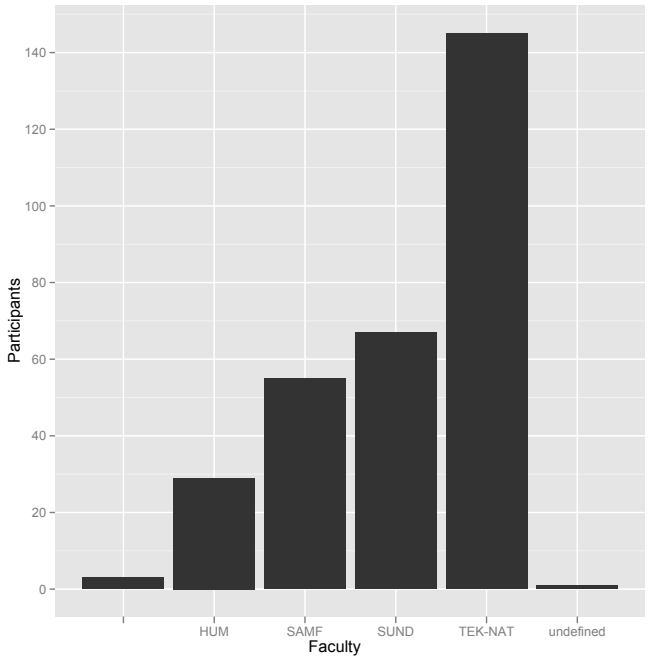



**Figure 4.1:** The distribution between the number of participants from each faculty at Aalborg University. The total number of test participants reached 300 in the period from the 28th of September to the 5th of October 2011. The faculties are HUM - The Faculty of Humanities, SAMF - The Faculty of Social Sciences, TEK-NAT - The Faculty of Engineering and Science, SUND - The Faculty of Medicine, SBI - The Danish Building Research Institute.

Initially, the distribution of test subjects in the dataset between faculties at Aalborg University are plotted. As seen in Figure 4.1, almost half the participants of the total 300, are from the TEK-NAT - The Faculty of Engineering and Science.

Prior to running the test - an initial test was conducted to ensure that no test results would be lost due to unnecessary hassles and point out optimization issues that could improve the high fidelity prototype and the acquired data.

## 4.1 Initial Iteration

Prior to testing the chosen demographic, the prototype itself must be checked for inconsistencies and potential pitfalls that can end up biasing the results due to malfunctions

or distinct problems with the interface such as communication of task descriptions, purpose, legal, design issues, etc. Therefore, the initial iteration of the development model is focused on producing a prototype that can be evaluated by selected representatives from different faculties at the university prior to testing on the selected subgroup of intended test subjects. This resulted in some surprising and helpful responses from the respondents:

- A short description is needed so the user knows the purpose of the tasks even though they are meant to be abstract.
- Users from faculties of economics and social studies tend to be more impatient if the test are long, compared to users from the technical science faculties - who tend to follow the tasks even though the purpose is not clear.

This initial beta-test of the test led to design changes such as the ability to leave the test at any time if it is deemed too obnoxious, tedious, or outright boring to the users. Additionally, the demographic information had to be filled in prior to testing the users in the constructed environment in order to provide them the opportunity to quit the test prematurely and still collect the information necessary for the dataset, because a feature about the overall time the test subjects would be willing to spend on the tasks itself, could also be a feature in the test of the user behavior - as of the results of the initial test listed in 4.1.

## 4.2 Investigations

The test investigations will be stated such that it will be possible to detect difference in behavior due to the distraction types introduced in Subsection 2.2.2 or the interaction id, i.e. number in the sequence of each task, the response times, and the accuracy will be the dependent parameters of interest in this regard. The differences in user behavior will be investigated using the following 5 investigations:

1. If there is a difference between the response times in between tasks of the test subjects in regard to the distraction category in the test.
2. If there is a difference in the response times in between tasks of the test subjects in regard to the interaction sequence in the test.
3. If there is a difference in response time between the category of distraction and the type of task
4. If there is a difference in the ability of the user to complete the tasks with regard to the category of distraction.

5. If there is a difference between response time and ability for each task - an increase in errors in regard to the type of interaction id in the test.

### 4.2.1 Statistical Tools

The tools for segmenting the acquired data from the test have been chosen to be the R statistical language and environment [R Development Core Team, 2011] along with additional packages. It is free and has the core functionality necessary to access the database, convert JSON [Lang, 2011] into its own variables, and has additional packages such as reshape [Wickham & Hadley, 2007] and ggplot2 [Wickham, 2009] to change the structure of the data so that it can be plotted. The data is directly accessible from R using the package for SQL queries [James & DebRoy, 2011].

The dataset can be acquired as a CSV-file on the DVD in the Appendix C along with the complete database SQL-dump, containing all recorded data. The data structure follows the implementation described in Section 3.2 - which is very important to understand when accessing the JSON data structures that logs the interaction activity. Consideration of such issues is not of any concern when working with the "," delimited, CSV-file dataset provided on the DVD.

### 4.2.2 Investigation 1

For each successful interaction, a set of data about the interaction is logged - this log confines to the data-structure defined in Figure 3.3. The log itself does not compute the response time but only logs the time from the start of the task. Therefore all entries pr. task must be subtracted from the time parameter of the previous entry.

$$dt = t_{entry} - t_{entry-1} \tag{4.1}$$

Each user now has a response time for all 10 entries in each task, for each task type, and each distraction, meaning we have a feature space for each interaction, which renders at a total of $10 * 8 * 4 = 320$ entries at maximum for each user according to 2.1. All complete entries are then merged into one dataset with its response time values, ids for each entry in the sequence, and the user average response time. The data is plotted using facetted histograms available in the ggplot [Wickham, 2009]package for R - each facet represents a distraction type. The binwidth for the histogram will be set to 1, which implies that each bin covers 1 ms. - the resolution of the data is 1 ms.

### 4.2.3 Investigation 2

The parameter for this investigation is calculated in the very same way as for the Investigation 1 response time 4.1. However, the difference is that the distribution of response time

is investigated by interaction id, meaning the interaction sequence for each type of task, for each category of distractions. The time difference/derivative are stored in the dataset in the *dt* field.

### 4.2.4 Investigation 3

This investigation is based on the response time between each task from Equation 4.1 and the error rate derived in Equations 4.2 & 4.3. The tasks are plotted against the type of task, visual layout, that the user is exposed to.

### 4.2.5 Investigation 4

For each interaction a binary variable is determined for the correctness of the input in the sequence - the error parameter is therefore either true or false. The parameter is calculated for each successful interaction the user makes.

$$x = targetObjectValue - registeredObjectValue \tag{4.2}$$

Using the $x$ calculated in Equation 4.2 as a parameter $x$ to Equation 4.3 , a binary parameter is used to determine whether the interaction was correct or not, such that an error value, which is TRUE, represents an interaction that missed the intended target.

$$error = f(x) \begin{cases} FALSE & \text{if } x == 0 \\ TRUE & \text{if } x! = 0 \end{cases} \tag{4.3}$$

The errors are stored in the dataset in the *error* field.

### 4.2.6 Investigation 5

This investigation is based on the response times for each interaction compared to the type of task for each category. The type of task is plotted against the $dt$, computed in Equation 4.1, a linear regression of the response times for each investigation by category is super imposed to see if there is any tendencies in the response time by category.

## 4.3 Findings

The results of the evaluation according to the investigations introduced in Section 4.2, will be presented in this section. The findings of the investigations 4.1 will be explained in ascending order as follows in Section 4.2.2.

### 4.3.1 Explanation of variables

The 4 different types of distractions are named as *StrategiesTask* - which has no distraction in the environment, *MessyTask* - which introduces additional objects in the screen during interaction, *SizeTask* - which is exactly as *StrategiesTask* but communicates its order in the sequence by scaling the object according to its value instead of numbering & finally *2OrderTask / OrderTask* - which is a 2. order task that introduces new objects into the screen at random between each interaction without using a visual layout strategy.

- StrategiesTask - has no distractions in the environment.
- MessyTask - increases the number of visual objects in the scene with $k^2$ pr. object value
- SizeTask - no distractions in the environment, the visual object changes in size according to the object value
- 2OrderTask / OrderTask - introduces new object at a random position in the interface with an id incremental to the number of objects in the environment.

The investigations will all be interpreted visually and discussed briefly, if there are any interesting findings in the investigations they will be further examined in regard to the problem statement 1.7, The emphasis will be on differences in user behavior in the investigations, with regard to two independent variables response time and errors for each task by the distractions in the environment and the visual layout or designs that the test participants have been exposed to.

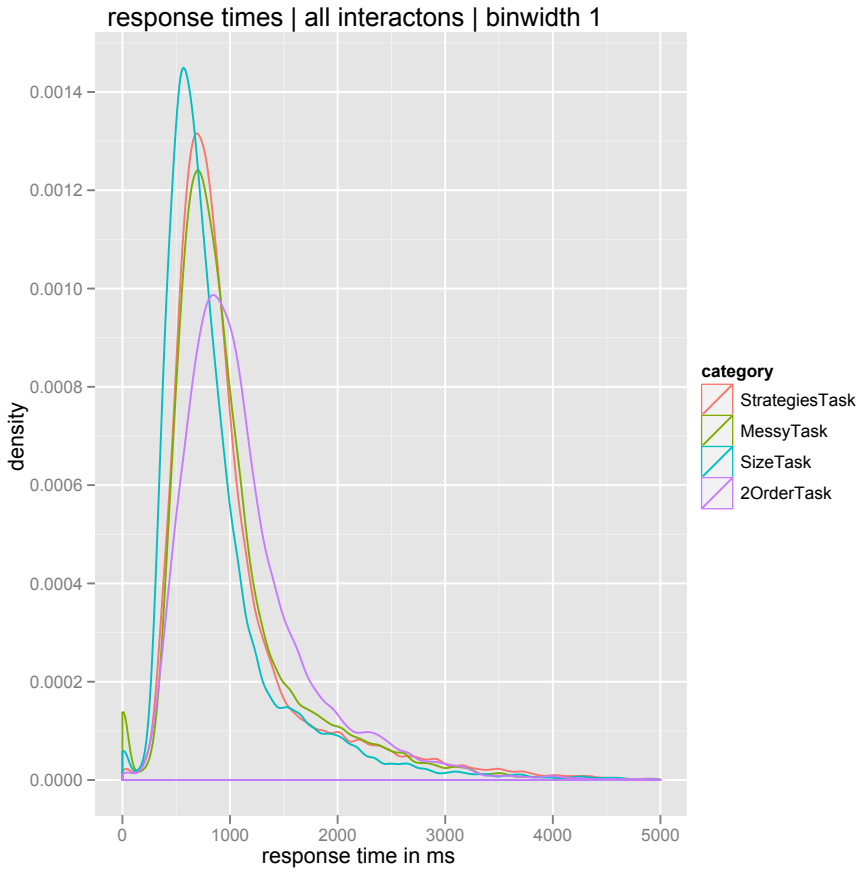



**Figure 4.2:** Investigation 1 - The density distribution of the response time of all valid object interactions, i.e. those who are registered in the database. There is a difference in the head for each distraction type, which should be investigated further, where as the tail events seems more scattered. See 4.3.1 for explanation of variables.

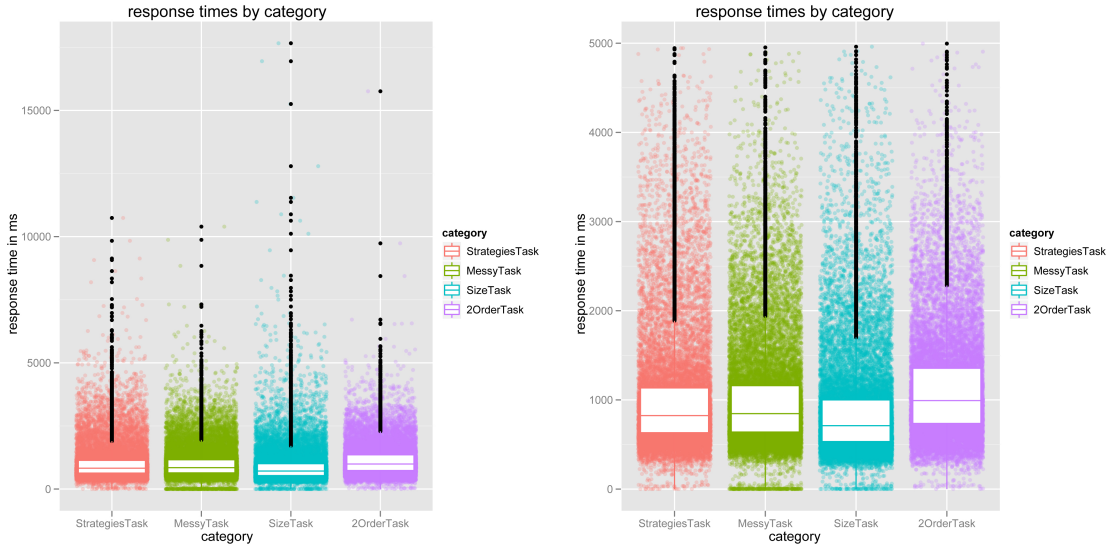



**Figure 4.3:** Investigation 1 - The box plot further shows the difference in average response times between the type of distractions in the interface. See 4.3.1 for explanation of variables. Zooming into the range of *0-5000* ms, we find a slight, but not certain, difference between the categories of distractions.


**Figure 4.4:** Investigation 1 - Density plot & histogram of the head distribution of all response times in the range of *0-2500* ms. There are differences between each task categories' response times. See 4.3.1 for explanation of variables.

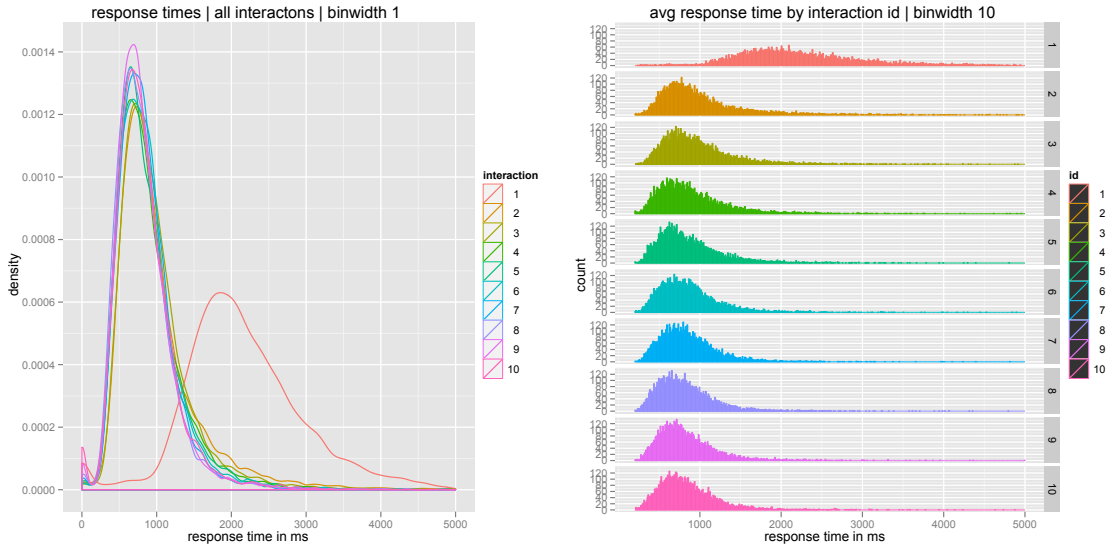



**Figure 4.5:** Investigation 2 - The density plot & histogram of all interactions by ID, notice that only interaction *id 1* distinguishes itself from the other interactions, *id 1* being the initial interaction during each task of a total of 10 possible.

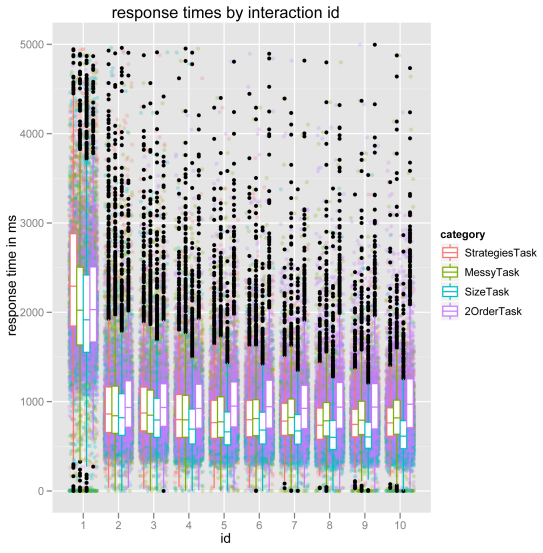



**Figure 4.6:** Investigation 2 - The response time pr. interaction id plot, shows a difference between interaction id 1 and the rest of the interactions. However, there does not seem to be significance in the difference between any of the interactions. See 4.3.1 for explanation of variables.

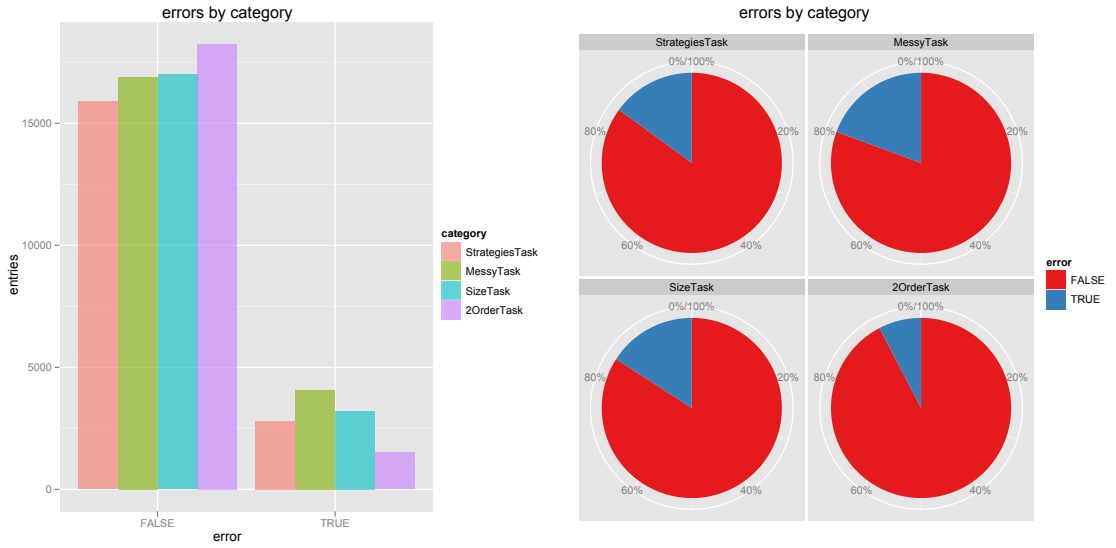



**Figure 4.7:** Investigation 3 - The error rate by category of distraction. The total number entries are 79609, in relation to the y-axis on the graph & the percentage wise error rate by category of distraction. Error = FALSE indicates a succesful completion of the task by the participant.

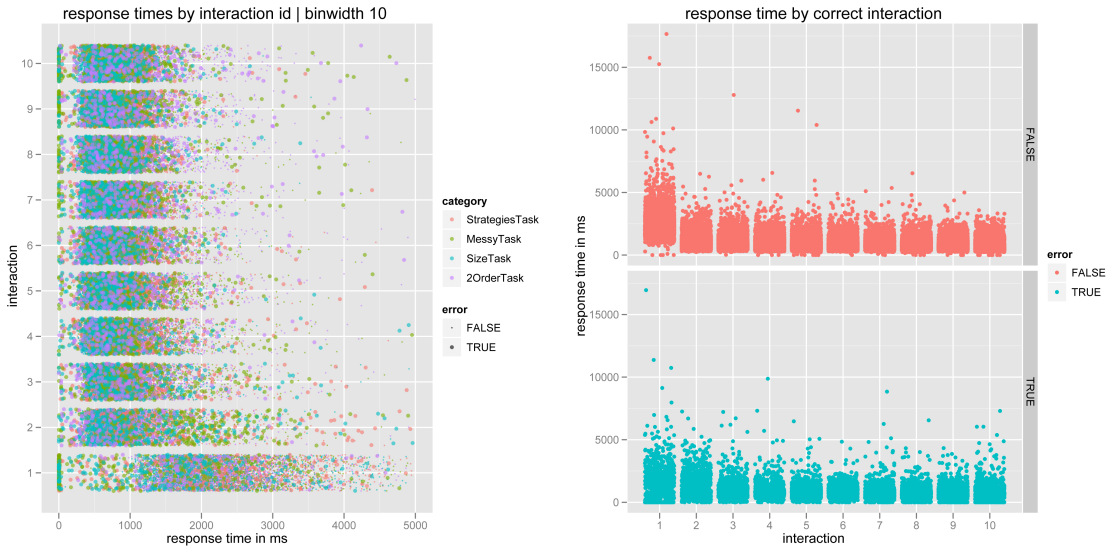



**Figure 4.8:** Investigation 4 - The error rate by category of distraction. The total number entries are 79609, in relation to the y-axis on the graph. Notice the pattern that correct responses, where error = FALSE, have longer response times than entries with an error=TRUE.

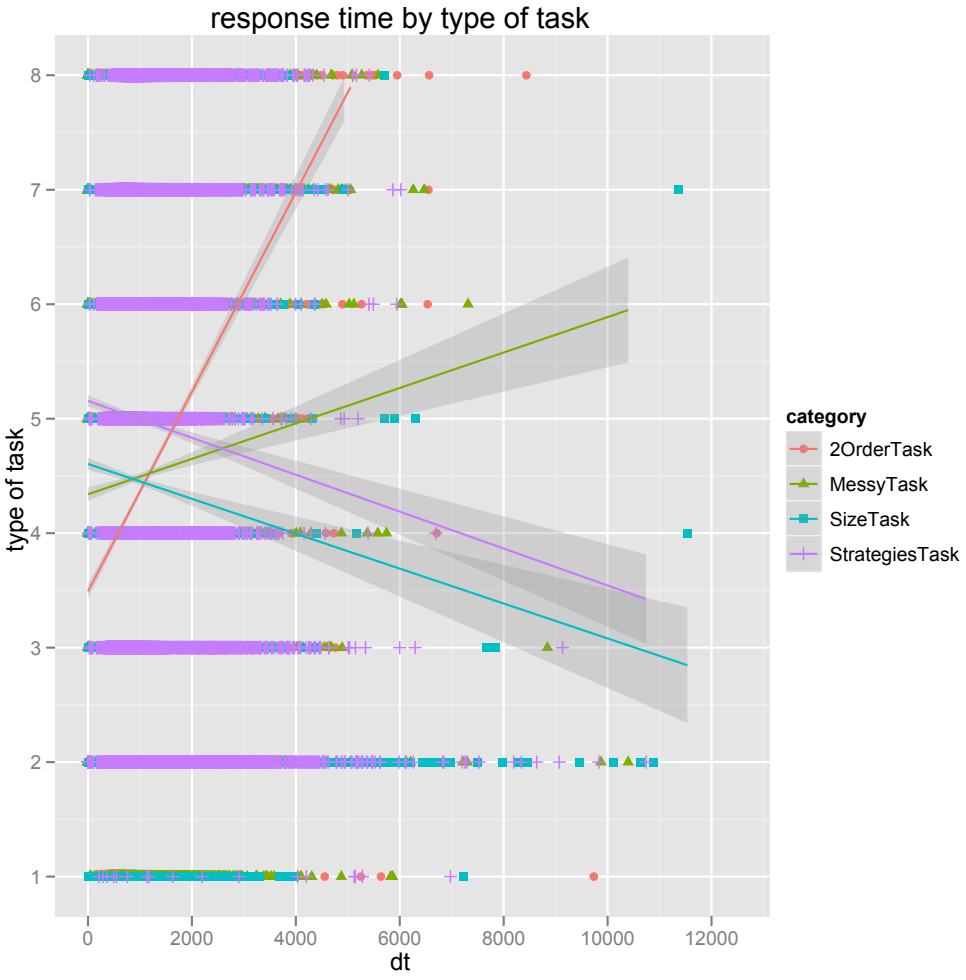



**Figure 4.9:** Investigation 5 - The response times compared to the type of task for each category. The regression lines show that the response times for SizeTask & StrategiesTask are decreasing for each type of task while the response times MessyTask and especially 2OrderTask are increasing for each type of task - which indicates that these two pairs should show a *difference* that could be *significant*.

Findings in the investigation described in the beginning of the chapter 4.2, where possible findings will be reasoned for and possible causalities discussed briefly. Each of the following investigations will try to establish relationship between the user behavior and user ability - findings that indicate a difference in user behavior will be elaborated further.

*Investigation 1* indicates that the distraction type influences the response time for each task. Though, these influences cannot be completely distinguished, but the tendency is present. The SizeTask, as seen in Figure 4.4 shows an indication that indexing happens faster than in the categories where object representation is numerical by character. This indicates that the size of an object can be a fast way of signaling relevance to the user, hence there is a measurable effect on the response time for a given user.

*Investigation 2* indicates that there is a visible difference between response time of interaction id 1, and the other 9 response times. However, the superimposed box plot in Figure 4.6 cannot be completely distinguished, but the plot indicates that a difference between the 1. interaction and the other 9 can be detected with marginal errors. The indication that the response time is dependent on the interaction sequence can however possibly be explained by the initial animations that configure the task layout and therefore are only occurring in the test at the onset of the first interaction in each task. The time is logged from the onset of the task and not at the finish of the animation, making the first interaction, id 1, stick out from the other response times. This issue could bias the results so much that it could be responsible for the 1000 ms approx. difference that is present in the response times. Because there is animation of the new positions in the visual layout in the 1. interaction a time error is introduced. Therefore we cannot with certainty assume anything about the difference between that interaction and the other 9.

*Investigation 3* indicates that the category of distraction influences the ability to answer correctly. Looking at both plots in Figure 4.7, there is a difference in the number of correct answers according to the category differences in the interaction environment. The difference in overall errors by category of task ranges between 5% for the 2OrderTask & 20 % for the MessyTask, which can be explained by the number possibilities for errors are less in the 2OrderTask than in the MessyTask since fever objects are shown.

*Investigation 4* indicates that errors are more likely to occur, the shorter the response time, as seen in Figure 4.8 - especially in the case of the initial interaction in a given task. The error parameter's targetID increases for each interaction - which means that if a user interacts with object 2 when his target is 1, then the next target object is 2 no matter the success of the previous action. Object 2 will now be marked visually as already visited to the user, even if it is the next correct object to interact with, which could confuse the user. These errors are not accounted for in the treatment of the variables and could be a source of error as well.

*Investigation 5* indicates that users can make a strategy when the sequence of tasks is visible to them, as with SizeTask & StrategiesTask, where the response times are decreasing

for each type of task, while MessyTask with an increasing number of distractions and especially 2OrderTask, where the full sequence are not shown, are increasing in response times for each type of task.

These investigations indicates that there is a difference in response time for each task interaction with the interface. *Investigation 2* indicates that the initial interaction response time is greater than the subsequent - but this finding can be errorprone because there is animation between the type of tasks which is included in the response time - this source of error makes eventual findings in regards to interaction response times very hard support. That coupled along with the findings of *Investigation 3 & 4* indicates that errors depend on the category of distraction or the ability to make errors, and more interestingly that errors are more likely to occur the shorter the response time.

The findings of *Investigation 5* show an interesting trend in the linear regression models for each category of distraction, which indicates that changing the environment can affect the response times. There are two trends represented by two pairs of categories (SizeTask, StrategiesTask) and the (2OrderTask, MessyTask) pair. The trends have different signs on the slope of the linear regression line which indicates that the (SizeTask, StrategiesTask) has longer response times as the test proceeds whereas the (2OrderTask, MessyTask) indicates that the response times become longer as the test proceeds. This relationship should be verified by a further statistical analysis - which hopefully will provide an answer to the Problem Statement 1.7.

## 4.4 Results

The linear regressions in Figure 4.9 shows tendencies that the category and type of task has differences with regard to their response times. The type of category seem to have 2 different slopes - the tasks with a visual layout strategies all have shorter response times than the 2OrderTask. The increase in response times in the 2OrderTask can be explained by the fact that the sequence of the task is not revealed and thereby it does not decrease in difficulty as it progresses which seems to be the case in the ordered visual layouts.

This visible difference should be further investigated to ensure that the *difference* is *significant*. The mean values of the response times will be investigated for all users, because the investigation parameters are to find *differences* in the response times between the different tasks, not associations of relationships between the tasks. The approach will be an Analysis Of Variance as described in [LeBlanc, 2004, chapter 11, 13], that will provide further legitimacy to the finding of *difference* in the mean values of the response time, the dependent variable, type of task, visual layout that the user encounters, and the category of distraction in the environment. In order to do so we must provide a working hypothesis in order to establish whether there is a *significant* difference in response times:

> $H_0$ - *There is no significant difference between the categories of distraction and the type of task that the user completes with regard to the response time for each interaction with a probability 4.4.*

The null hypothesis reflects the problem statement and furthermore results will be having a 95 % certainty of being true in regard to this experiment. This certainty is then reflected in the probability 4.4 that there is a difference in response time when changing the environment that the user operates or the visual layout in the tasks.

$$H_0 : p \leq 0.05 \tag{4.4}$$

This probability establishes our working hypothesis where we can apply ANOVA. ANOVA relies on 3 basic assumptions in order for the results to be valid:

- Randomized unbiased study design
- Population distributions are normal
- Population variances are equal

These premises will be validated in the following sections.

### 4.4.1 Study Design

Testing user behavior have relied on a set of assumptions about the experiment, described in Subsection 2.5, but to interpret the data another set of assumptions are necessary. The test subjects themselves are not what is tested in this experiment, it is their behavior - hence the user behavior is treated with 4 different categories of distractions. These are not assigned randomly because there will be no way of not treating the user's to establish a default user behavior. This is sought in the design of the tasks and the 2OrderTask will serve as the unstructured representation - control group whereas StrategiesTask, MessyTask & SizeTask will serve as the treatment representations of user behavior. There have been no random assignment of tasks hence they all have been presented to the user in the same sequence throughout the test. This is based on the assumption that the task of sorting the objects in ascending order are so simple that the noise, or errors, in the resulting responses are due to the difference in the treatment not due to user's not being able to understand or comprehend the task.

Because the initial interaction id has a mean that is different than the other interactions, see Figure 4.5 & 4.6, and that we want to find user behavior in accordance to the treatment group, not the general interaction with the interface - this interaction id will be removed. Linear regression models are very sensitive to outliers and the fits will be greatly influenced by this set of outliers in regard to user behavior. Furthermore, there seems to a pattern of the users to first asses the environment and then execute their plan, but since there has been used animation during this interaction we cannot with certainty use the initial interaction as an entry in the dataset to determine the difference between the treatment groups. The types of tasks will also be assessed as another independent variable in the experiment because the visual layout are the same in all the 3 categories of visually logical representations. Experiment assumptions about this ANOVA evaluation are:

| **Description** | **Variable** | **Identifier** |
|---|---|---|
| Distraction category | Treatment Group | category |
| Task type | Treatment | type |
| Response Time | Effect | dt |

### 4.4.2 Normality

For each of the groups the categories of distraction, the normality will be assessed. The task type have been the same for all and the normality is considered defined by the completion of all tasks which are the same for all subject in this subset of users. The mean of the boxplots in Figure 4.4 are evenly distributed with a slightly larger 3. quantile than the 1. quantile. This seems like a response pattern that could be found in sounds, etc., but also have carry similarity to a normal distribution in the histograms in Figure 4.4

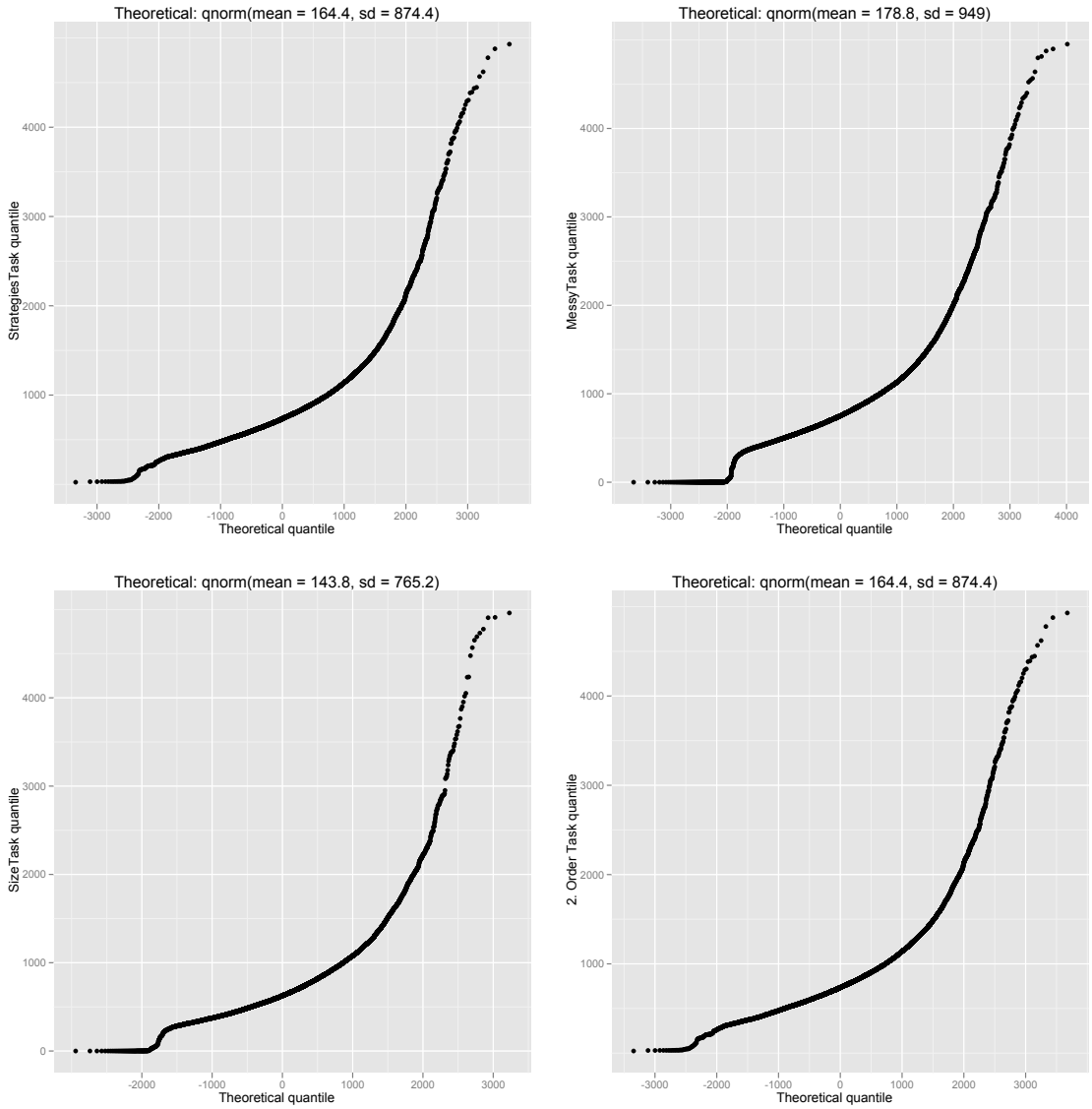



**Figure 4.10:** The quantile plots are not perfectly normally distributed, but will be deemed ok for the ANOVA test because the sturdiness of the model itself. All quantiles also assume the same shape which can be interpreted as that they all carry the same amount of error in them - and therefore should be comparable.

| Levene's Test for Homogeneity of Variance (center = mean) | | | | |
|---|---|---|---|---|
| | Df | F value | Pr(>F) | |
| group | 3 | 240.38 | < 2.2e-16 | *** |
| | 71578 | | | |
| Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 | | | | |

**Table 4.1:** The Levene test finds that homogeneity for the means of the response times from different categories of distraction that represent the treatment groups, are significant.

The quantile plots in Figure 4.10 are not perfectly normally distributed as was indicated by the box-plot quantiles. The trend is however similar to all groups and therefore the error will be distributed similarly and therefore should be comparable in the ANOVA model.

### 4.4.3 Homogeneity of Variances

The F-test determines the validity of the normality assumption by testing the ratio between the largest and smallest Sum of Squares. However, since we already know that the normality assumption will violate the F-test from the stretched assumption of normality, as seen in the quantile plots in Figure 4.10, a Levene test will be used [LeBlanc, 2004, p. 259], because it is less sensitive to violation of the normality assumption.

The Levene test verifies the homogeneity or equality of the normality assumption and therefore we will be able to justify the use the ANOVA model to test for differences between the response times in between the treatment group - in regard to the distraction categories and the task types.

### 4.4.4 ANOVA test results

The test relies on the above mentioned assumptions and there are two independent variables, *the distraction category & type of task* - visual layout for each test participant. The dependent variable is the response time.

The null-hypothesis is rejected by ANOVA with an effectively $p = 0$ probability that there is no *difference* between the distraction category and the type of task with regard to response time for each interaction. This result tells us nothing about what this *difference* is but an effect plot of the means of the response by type of task for the different categories of distraction might help to see that *difference*.

With a rejection of the null-hypothesis, $H_0$, we are presented with strong evidence that at least two of the population means are different. This presents the problem of finding what mean is differing. Figure 4.11 shows an effect plot of the response times.

| ANOVA Table (Type II tests) | | | | | |
|---|---|---|---|---|---|
| Response: dt | | | | | |
| | Sum Sq | Df | F value | Pr(>F) | |
| category | 6.7209e+08 | 3 | 1376.43 | < 2.2e-16 | *** |
| type | 1.3028e+09 | 7 | 1143.44 | < 2.2e-16 | *** |
| category:type | 3.4713e+08 | 21 | 101.56 | < 2.2e-16 | *** |
| Residuals | 1.1646e+10 | 71550 | | | |
| Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 | | | | | |

**Table 4.2:** The ANOVA finds that there is a significant *difference* in response times between the categories and the types of task. Hence the probability of the category, the type of task and their combinations is lower than the 0.05 probability threshold set in the null hypothesis, we can reject the null hypothesis $H_0$.

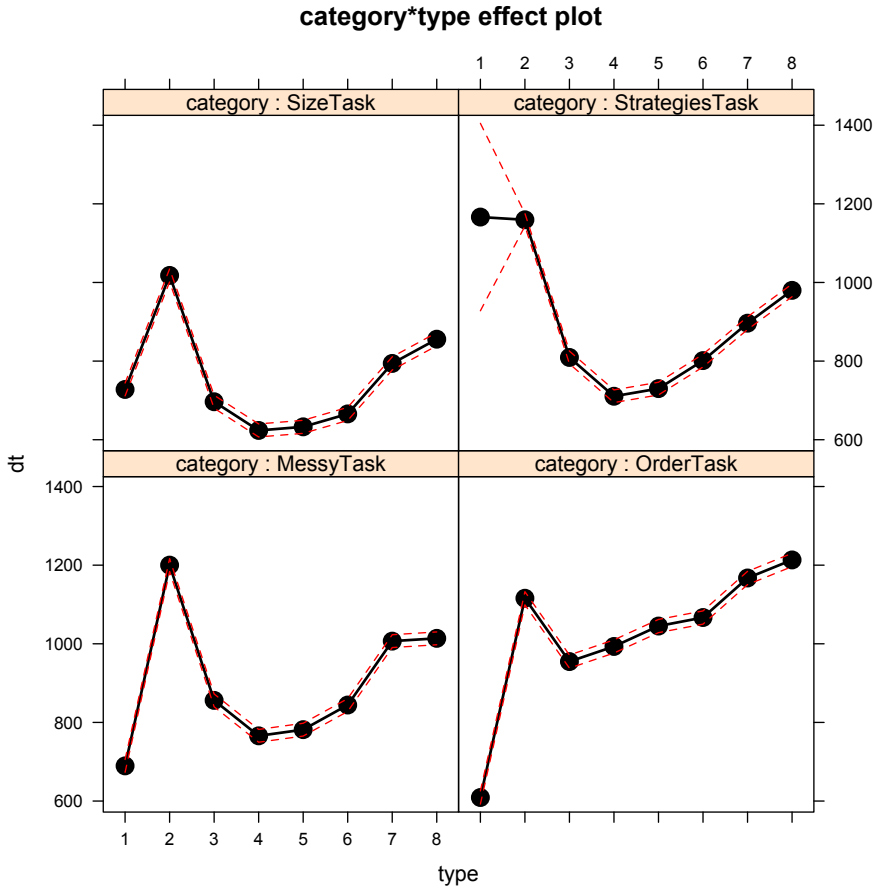



**Figure 4.11:** The effect plot shows that the response time between tasks in the order category are longer than in the categories with visual layouts. There is an anomaly in the StrategiesTask which cannot be explained immediately - but seems a possible source of error. However the rejection of the null hypothesis still stands due to the *significance* value.

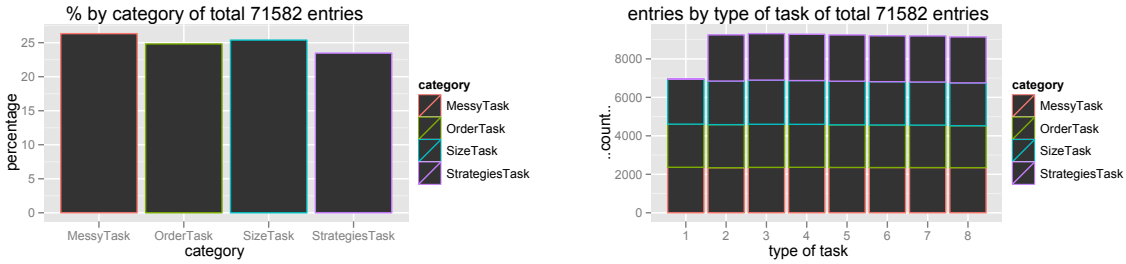



**Figure 4.12:** The sample sizes in the experiment show are not perfectly distributed. The left figure shows the percentage of number of samples by category of distraction which is distributed evenly with around a 2% difference. The right figure is the number of samples distributed by type of task and this shows us that there is hardly any samples that represents the first type of task in the StrategiesTask, in an otherwise evenly distributed set of samples. This source of error might explain the 2 % difference in the left figure.

### 4.4.5 Comparison of means

With the strong evidence from the ANOVA test that there is a difference in the means of the response times between categories and type of task - we need to investigate which of the means that differ. Looking at the sample sizes for each of the independent variables will be essential for choosing the right type of test for comparison of means

The distribution of samples shows an even, but not perfectly even, distribution of samples for each category of distraction and type of task. Looking at Figures 4.12 there are samples missing in the 1. type of task for the Strategies category. This is either a result of the logging mechanism not saving the entry into the database on the client-side or due to preprocessing errors in the scripts that created the dataset from the database. The offset in the distribution of samples by around 2 % can therefore be explained along with the strange artifact in the StrategiesTask effect plot in Figure 4.12, which cannot determine the slope of the initial task because of a lack of samples. This error will only influence the choice of test for comparison of means, to be a method of evaluating different sample sizes. Since we want to find which mean values that differ, it will be attempted to explain the variation in response times by *category* and by *type of task* separately. This will done to either verify whether the difference in means can be explained by the distractions in the environment or by the different layouts or designs of the of the type of task that the user has encountered. The pairwise comparison probability will continue to be $p \leq 0.05$ from the null hypothesis, meaning that a difference in the means will be with a 95% confidence level.

The R package has a pairwise comparison One-way Anova available which were used to produced The Tukey Honestly Significant Difference (THSD) test that is used because we have two or more variables to explain the difference in means with regard to the response

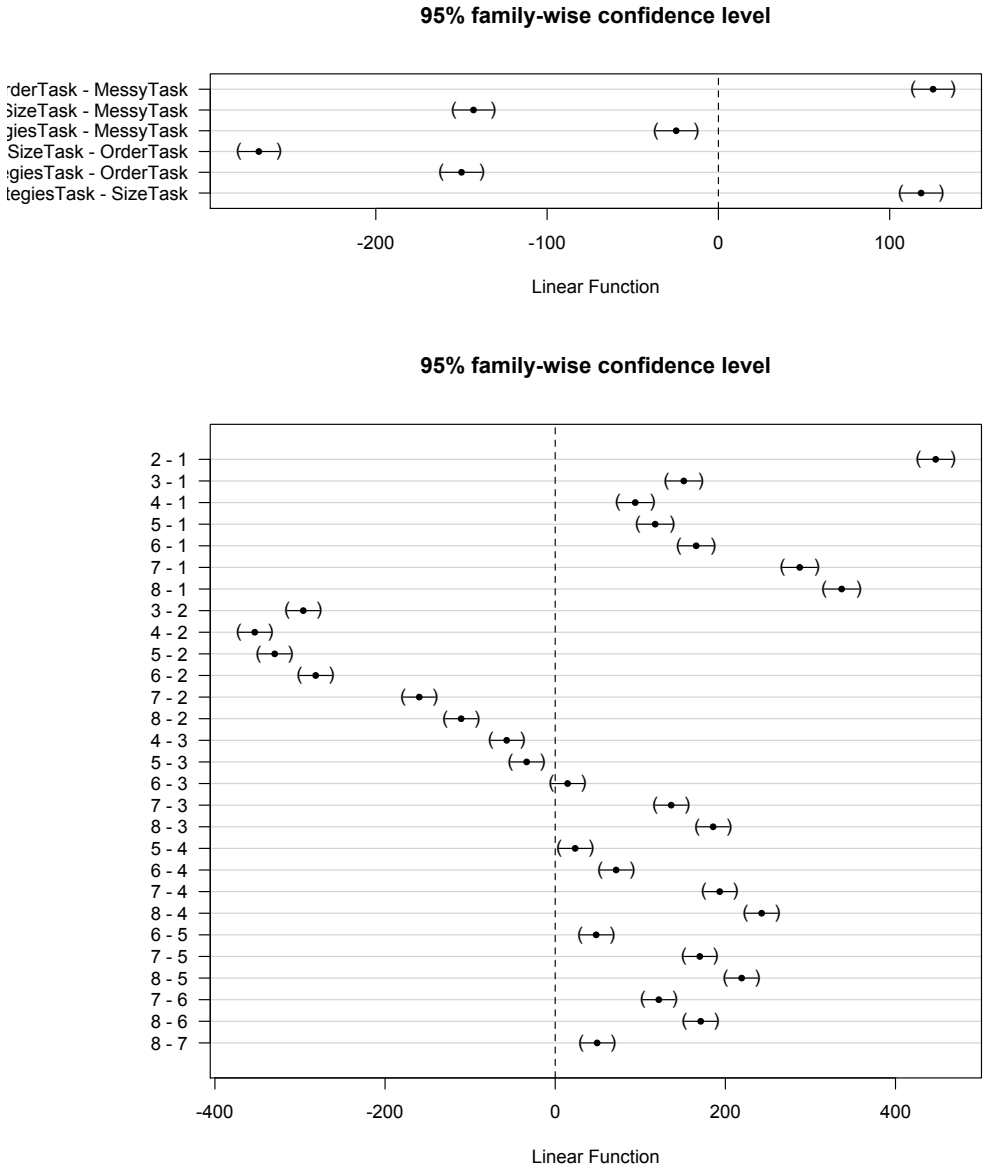



**Figure 4.13:** The Tukey HSD plots shows the pair wise difference between the means of the sample population means. The upper plot shows the difference in means between the categories of distraction, the only differing pairs are (SizeTask, MessyTask) & (SizeTask, OrderTask). The lower plot shows the same difference but for the task type where the only differing pairs are (2, 1) & (2, 7).

time - THSD test is also robust enough to be used with different sample sizes that we established was necessary. The THSD test, see Figure 4.13, shows how the means differ and the visual interpretation of the result is that the means differ at the $p \leq 0.05$ level in both Tukey plots, one for each independent variable. The pairwise distribution of means between response time and the categories shows that pairs (SizeTask, MessyTask) & (SizeTask, OrderTask) have *significant* different means. This leaves of with strong evidence not only that changing the environment, i.e. the categories of distraction, has an influence on the difference on the response time in user behavior. The types of task overlap so much that we cannot determine if one layout is better than the other - the mean difference in the layout types can be only be established with the pairs (2, 1) & (2, 7).

## 4.5 Part Conclusion - Evaluation

The findings in the investigations indicated that there was an increase in errors with regard to the category of distraction in the environment. This seems as a result of the possibility of making errors, that is provided by each of the categories of distraction - as shown in Investigation 3. The interesting findings of Investigation 5 shows that the category of distraction also has an influence on the response time which is investigated using ANOVA tests on a dataset without the initial interactions, interaction 1.

The results from the ANOVA test provides strong evidence that there is *significant difference* in the response times with regard to the category of distraction in the environment that the user is exposed to. Because the distractions are designed, we will not consider the *difference* is due to errors but rather the changes in the environment. Furthermore, the $p - value$ that rejected the hypothesis is effectively $p = 0$, with an $F - value$ larger than 1, we can establish that the difference is due to the changes in environment that the users are exposed to. The indications of the differences in the signs of the slopes of Figure 4.9 are thereby validated by our ANOVA test results.

The difference in regard to the type of task that the user is exposed to, does not show significantly differing means apart from the pairs (2, 1) & (2, 7). Looking at Figure 3.5 that describes the different types of layout strategies in the task - it seems that the cluttered layouts of 1 & 7 - are influencing the response time in comparison with layout 2. This relationship can be interpreted as a difference in how fast the user can complete the task presented to them in regard to the purpose of the task - putting a sequence in ascending order. The explanation for the difference in means can therefore be used to finding the best and the worst layout or designs in a set of different visual layouts - in this case the performance of task 2 seems the fastest whereas 1 and 7 can be considered the slowest with the rest distributing themselves in between.

The normality assumption is slightly violated, hence the distribution of samples are not perfectly normal. The amount of sources of noise in the test environment and the unknowns connected to the user behavior should have made influence normality of the response times - but the fact that the quantiles in Figure 4.10 are so similar in their deviance from normality, these sources of noise will be ignored and the Levene-test verifies that the normality assumption can be justified. The missing entries in StrategiesTask, as seen in Figure 4.12, explains the difference in the effect plot of the means 4.11 for the StrategiesTask where two different regression lines are proposed. This is not due to the samples that were obtained but the lack thereof. The *significance* probabilities does not seem affected by these missing samples and it might explain the difference in the sample sizes by category of distraction values but since the probability values are so significant even with this source of error, the conclusion of rejection of the null-hypothesis 4.4 with a 95 % confidence of there being no difference in response times is still considered valid.

# 5. Discussion

A high fidelity prototype has been implemented according to the specifications of the Design Chapter and the Development & Implementation Chapter and a dataset has been produced from testing a demographic of students at Aalborg University with 300 participants. The user behavior found in this test could reflect the general ability of such a demographic - which might be different if the test was conducted on a different demographic subgroup with different age, abilities, computer experience, level of abstraction, etc. than the participating students.

The use of different interfaces was expected by the researchers, but it turned out that only mouse and trackpad was used at large. This meant that other interfaces such as touch interfaces, slate PCs, and smartphones were not used. Since the test was distributed to the students' official university mail, it was probably received by the students at their working computers which could explain the lack of presence of the various other devices. This would also mean that the scale of the visual objects has not been a major parameter, which it could have been because the resolution on a smartphone today almost has the same resolution to the one seen in computer screens - while having significantly smaller displays.

The demographic of university students might also explain the very low response times for some users - they seem very capable of processing simple tasks, in the range of 3-500 ms. This is a very short response time for an action, which leaves almost no time for cognitive processing. The system is able to detect such differences so calibration could be useful for time critical user interfaces as well as representative interfaces such as described in this report.

None of the distraction types seemed to be too difficult to complete for the users, which would have caused them to reach their maximum capabilities of interaction with an interface. The individual limit of each user could prove to be distinguishable as a measurement of user behavior.

The method used to determine whether there is difference in the response time for each interaction in a sequence - produced robust results that reflected the user behavior during the simple tasks. The method used to verify the results of this project have been crude and exploratory - yet with rigorous argumentation and using the data to guide the way to the results - a significant result has been acquired.

# 6. Conclusion

This project has focused on the aspect of the threshold between a hidden and a perceived affordance of a given object in an interface as an expression of user behavior - the term *meta-affordance* is introduced to accommodate for a relationship between data. Investigations were made to clarify whether user behavior can be detected by measuring the performance of users solving simple tasks in an interface. The test itself was designed to reveal differences in user behavior - more specifically the response time between interactions and their ability to sort a visually dispersed set of objects. The distractions were introduced in the interface to compromise "optimal" behavior so that the effects of these distractions could be observed in later data processing.

The results show that there are indications of common user behavior, which reflects the environment and the visual objects ability to communicate the *meta-affordance* state - hidden or percieved. The shorter response times in the SizeTask indicates that the size of a visual object is a good indicator of a relationship as a *meta-affordance*. It also seems as a valid indication that the ability of each user, i.e. their error rate as a parameter for the affordance state, hidden or perceived, of a given object in any configuration could be a valid approach. The difference between the initial interaction and the rest of the ids seems as much an artifact of the implementation than that of the user behavior, which indicates that such results are a combination of user behavior, environment and the visual layout/design of an interface. Hidden *meta-affordances* can therefore be identified and omitted for a user who cannot detect these available affordances. The ability parameter can hence be directly mapped to a perceived affordance in our modified definition of *meta-affordances*.

The response time for each *meta-affordance* which are perceptible to the user, can be used to describe the affordance or its quality - i.e. the faster the response time the better, as shown with the SizeTask - that indicates its relationship to other objects by the visual objects size. Indications are observed that the size of a virtual object is a better representation than a numerical character, since the response times of task where the interactions were performed on objects varying in size, was generally faster. This feature could be interesting as a quantitative description of the affordance between a user and the interface they are using.

The distraction categories in the environment does not immediately show any difference, hence they haven't been distractive enough for the users to not complete the tasks, however from the analysis of means in the response times show a different story. The results from testing the response times against the categories of distraction implicates that the environment that a user is presented when interacting can be differentiated. The difference is significant between the categories of distraction in the environment presented to the user which serves as an evaluation of the different environments. This common relationship between response times and the environment indicates that a method of calibrating user behavior generically is possible. Such a calibration method does not necessarily need to environmental distractions embedded in the interface but could simply be calibrated by

the user in different settings.

Since the response times for each id subsequent to id 1 are highly consistent, it seems as though the interactions expresses the individual users' optimal performance in this particular interface. If the difference between id 1 and the subsequent ids could be accounted for without the use of animations prior to task initialization, could potentially provide a state estimation of the users behavior - such as *planning* or *execution* of a task. More test results would be required and such a finding will note be possible with the dataset acquired in this project. Though, even when removing the idle time gap, the response times for id 1 still has a greater variance compared to the rest, which could indicate that users differ in how they plan/how long they are planning executing a series of tasks.

It can therefore be concluded that *"it is possible to detect differences in the user behavior, while performing simple tasks in an interface, changing only the complexity of the environment"*, as asked in the Problem Statement 1.7. The results are based on only a single parameter, the response time, that produces *significantly difference* in the tendencies of the users' behavior when interacting with an interface.

# 7. Perspective

The results of the evaluation, shows that user behavior can be modelled in relation to a given interface. In this project a hifi - prototype that was unknown to all test subjects. The test subjects overall have performed the test in a very similar way - which both reflects the interface was succesful in communicating the tasks that the users have completed - and that the interaction response times and error rate in a simple repetetitive task can show trends in user behavior. The user behavior is dependent on the interface, so are the parameters of response time and ability in the given unknown environment. The results could also be interpreted as an indicator of the interface ability to be used - its usability, by the user as a a feature representing variables that depends on the mixed contributions from user behavior, user environment & the application itself, etc as a whole without extensive modelling of the complete environment that the user is exposed to at any given time - opening for quantifiable evaluation of an interface in regard to its users, its environment & the task it seeks solve. This reaffirmation between the user and the interface could prove a very interesting method of diagnosing and quantifying the quality of a given user interface *meta-affordances* with regard to the user behavior. The method of evaluation used in this project could therefore be evalutated in any interface In its *meta-affordance* as being perceived or hidden by a similar test

One could expect affordance restrictions in a design guide validated qualitatively on the design-wise validation of *meta-affordances* vs test values for a given demographic. This could automate validation of user interfaces very much like unit-test drives software development today. Using a similar method to quantifiably describe user interaction with *meta-affordances* in other environment with a different demographic group than this project could prove very interesting, because it provide developers with a much needed standard for testing interfaces according to the users current behavior or behavior in environments that are not known. *Meta-affordance* usability - testing could in that regard be an interesting and valuable tool for developers, designers, managers alike.

Using a generic method of user calibration, as shown possible in this report, could be used to produce a dataset with a demographic benchmark for specific age groups - mapping how they interact with all visual elements in an api - which could serve as an evaluation tool for interface designers. The averages of ability and response time could be used as a weight for an Artificial Neural Network. This "response" ANN could then be applied to emulate or aid the user behavior - resulting either in interfaces that adapts to the users dynamically, or in test environments where specific user behavior is emulated in designs or environments - validating the them in an automatic way. The task that the user completes must be simple to with regard to *response time* that is used to describe behavior in this project - other features could refine the description of the user behavior, ie. mouse movement patterns. A generic description of user behavior could also be used to refine the User Models used in Adaptive Hypermedia Systems or the User Models used for personalization which is used extensively in current online services ie. google & facebook.

# List of Figures

# A. Experimental setup

### A.0.1 Prototyping High vs Low Fidelity approach

Because detection of user requirements by simply asking them might be a difficult assignment (users tend to tell what they *think* they want, not what is possible), the need for designing a prototype, by which the users can experience pros and cons of the chosen issue, arises. Also, the actual use of a product (in the early stage), can expose requirements that neither the users nor the developers have thought of, because using the product might trigger situations and new application methods, which could not be foreseen in the initial design process. Using the ISO 13407 development model can introduce such findings into the development process in this case a well designed prototype can help to detect requirements, restrictions, etc. early in the process. A low-fidelity prototype made of simple materials, e.g. paper cards representing user screens, is most often preferred in the early stages for proof-of-concept, because any necessary changes to the system are not limited by the need for preserving already accomplished work. Liddle (1996) advises conduction of prototyping prior to any development [Sharp *et al.*, 2007, p. 530]. This way, the development team does not risk loosing all the work, if the system has to be totally redesigned.

While the low-fidelity prototype might be the preferred method in many contexts, it also has some clear limitations; in the specific case of measuring whether or not different subgroups of users might react different in designed environment, an analogue representation of the environment will be insufficient due to the following reasons:

- Time measurement inconsistencies.
- Limited error checking as the users complete the task
- Limitations in navigation and flow of the application, hence the interaction is imagined
- Facilitator-driven and requires a lot of manhours to complete in a consistent manner.

The errors of the prototype test will be the frame of reference for detecting any differences between the subgroups of users and therefore the error logging is essential. The tasks for the test will be related to navigation in the environment, which will not be comparable to an analogue low-fidelity representation. Finally, a paper representation of the system will depend on the actions of a facilitator, who then will serve as a possible source of errors.

In this project, where a test will be conducted to determine whether or not specific user behavior in a given environment can be detected, a high fidelity prototype will be used, since this will be embedded in the eventual deployment environment and should therefore provide the team with a consistent environment to test the proposed method. This way, a test conducted in a non-similar environment, would not show the necessary results in order to prove or reject the hypotheses specified for this project.

### A.0.2 Client-side Evaluation Approaches

The focus in HCI (Human Computer Interaction) studies have, until recently been in thorough laboratory practices to achieve consistent and reproducible results. This has removed the more industrious needs of cost-efficiency in the development process. Giving developers the tools and methods necessary to develop user friendly applications, by involving the users early in the development project can, according to research, be useful in eliminating critical errors in an application development process.

> *There is a need to reduce the implementation time in order to be able to spend more time on the evaluation of the prototypes. To reduce the implementation time and to improve the interface's friendliness, the literature approves the need for software tools...* [Leichtenstern *et al.*, 2010, p. 317]

Because user evaluation in application development is costly, there is a need to differentiate the methods applied in regard to the cost efficiency of the approach. This has been pointed out in several recent publications, such as [Leichtenstern & André, 2010] and [Leichtenstern *et al.*, 2010], which provides leverage to make laboratory tests more cost efficient and less artificial, by using interaction methods and approaches that is focused on field study approaches. The conclusion is, however, disappointing from a cost-perspective point-of-view, because a thorough understanding of a user interface quality still heavily relies on laboratory experiments, but the amount of time spent on evaluation in a laboratory setting can be reduced by using remote evaluation in the development process. The findings show that the use of laboratory experiments can be reduced to the end of the development life-cycle.

> *Overall, the results showed that field tests cannot be completely substituted by laboratory studies and should be used at least at the end of the development process to investigate specific user behaviour in different contextual settings.* [Leichtenstern *et al.*, 2010, p. 316]

The laboratory test procedures early in a design/development process can therefore be substituted by more cost efficient processes, as described by [Bruun *et al.*, 2009], which will be elaborated further in Section 2.2.3.

### A.0.3 Current commercial application development

Current requirements to online applications in the Web 2.0 paradigm means that applications can no longer be tested in isolated and simulated environments, because the affordances only seem relevant in a personal relevant context, i.e. social network applications, where the application itself requires online web services to function according to

their intent. Developing applications in a live environment is becoming the norm in web development, because the bugs involved cannot be replicated in a simulated environment.

> *Continuous deployments. People believe that if you go slower you'll get a better outcome? You can fix the bugs. But that's not true. The slower you go, the bigger the batch size and the more things go wrong. What if customers don't want your product? Do you want to find that out after you've built the whole product or only a tiny sliver of it?* [Adler, 2011]

The stretch between a simulated environment and the environment where the application is deployed is becoming greater, raising the complexity of implementation and of course the test approach.

Evaluation is key when developing applications, in an iterative manner, it requires significant resources to conduct the evaluation process. Reducing this cost and developing in a live environment are both aims of this project. The focus on resources in an industry based upon users, which most Web 2.0 applications/social websites are, there is an increasing need to conduct research in methodology, proposing innovative methods that can leverage this problem field, and make it accessible to developers which will eventually benefit the users. There are logistical considerations behind most application development strategies and a focus on cost reduction in the development process is of interest from the industry. A thorough tested and verified application is an essential interest to all involved in the process, but it can easily become too costly, hence there is a tendency to cut back on evaluation of an application from a management point of view.

> *Software organizations that develop and evaluate products for global markets or practice outsourcing or global software development face different but equally significant obstacles. When developers, evaluators and users are distributed across different organizations, countries and time zones, user-based usability testing, in particular planning and setting up the test, becomes a nearly insurmountable logistic challenge* [Bruun *et al.*, 2009, p. 1619].

The problem of cost efficient user evaluation is thoroughly described in [Bruun *et al.*, 2009], and can be optimized using an asynchronous approach that does not require direct involvement from developers/supervisors/experts, etc. The approach will be described along with an additional proposed method that extends the models proposed by [Bruun *et al.*, 2009], which will be presented later in this section. The test environments proposed in this paper evaluates 4 different user testing methods. The approach proposed by the authors is to have users evaluate a mail application using 3 different test environments, *UCI, Forum, Diary* that are less labour intensive than the traditional *lab* testing procedures using experts in the field as the control group [Bruun *et al.*, 2009, p. 1621].

- *Conventional user-based laboratory test (Lab) - Control Group*
- *User-reported critical incident (UCI)*
- *Forum-based online reporting and discussion (Forum)*
- *Diary-based longitudinal user reporting (Diary)*

[Bruun *et al.*, 2009, p. 1621]

The tasks the users had to complete was identical, but modified in accordance to the the context of the test environment. The tasks the users had to undertake in all test environments was to:

1. *Create a new email account (data provided)*
2. *Check the number of new emails in the inbox of this account*
3. *Create a folder with a name (provided) and make a mail filter that automatically moves emails that has the folder name in the subject line into this folder.*
4. *Run the mail filter just made on the emails that were in the inbox and determine the number of emails in the folder*
5. *Create a contact (data provided)*
6. *Create a contact based on an email received from a person (name provided)*
7. *Activate the spam filter (settings provided)*
8. *Find suspicious emails in the inbox, mark them as spam and check if they were automatically deleted*
9. *Find an email in the inbox (specified by subject line contents), mark it with a label (provided) and note what happened*

[Bruun *et al.*, 2009, p. 1621]

The tasks all served as a basis toward finding the discrepancy between the proposed test procedures. The parameters of interest was completion time and detection of problems. The *Forum* & *UCI* approach proved very time efficient according to Table 4 in [Bruun *et al.*, 2009, p. 1623] with findings of error of 24% & 21% of the total number of problems respectively. A Fisher analysis of the test results and time consumption reveals that these two approaches has an extremely significant difference to the *Lab* findings according to Table 6 in [Bruun *et al.*, 2009, p. 1623]. The UCI method - even detected all the critical issues that the *Lab* procedure also found. This finding is extremely interesting in regard to time expenditure in user testing, especially if early evaluation of application are of interest.

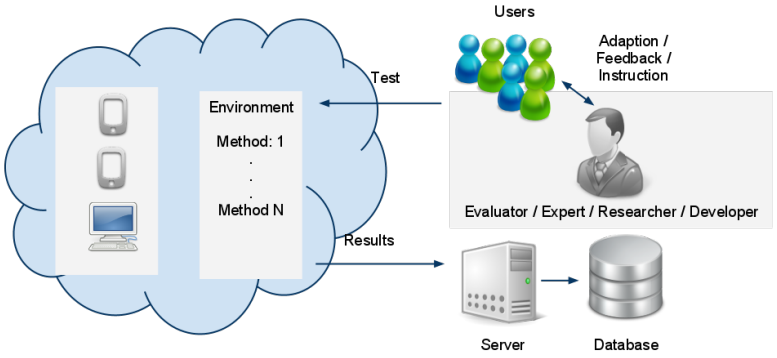



**Figure A.1:** The lab test approach involves synchronous involvement of both users and a supervisor physcially present who can guide the *users*/test subjects - during the test process.

In the following subsections will be described 3 methods that are chosen in order from least to most cost-efficient, starting with laboratory experiments, which has been the prevailing method of evaluation. The new constraints put on developers to develop integrated applications, as an example in social networks, makes evaluation methods that are asynchronous and not involving the developer a key development tool in order to establish a better relationship between users and developers, and of course make more relevant and better applications. The current memes in online development are described in the following subsections inspired by [Bruun *et al.*, 2009].

### A.0.4 Classic Laboratory experiments

Laboratory experiments involves a lot of labour and man hours in the research process and can be costly a affair. The approach on the other hand, provides the most rigorous feedback to developers and it is an essential tool in development of an application. As mentioned by [Leichtenstern *et al.*, 2010], it can be left until the end of the development life-cycle to deploy such a method.

The approach visualized in Figure 2.10, requires a supervisor to be present at all times during testing. Depending on approach and number of test subjects, this can be a tedious affair involving a lot of effort both during and after the test of the subjects depending on the choice of evaluation method prior to testing.

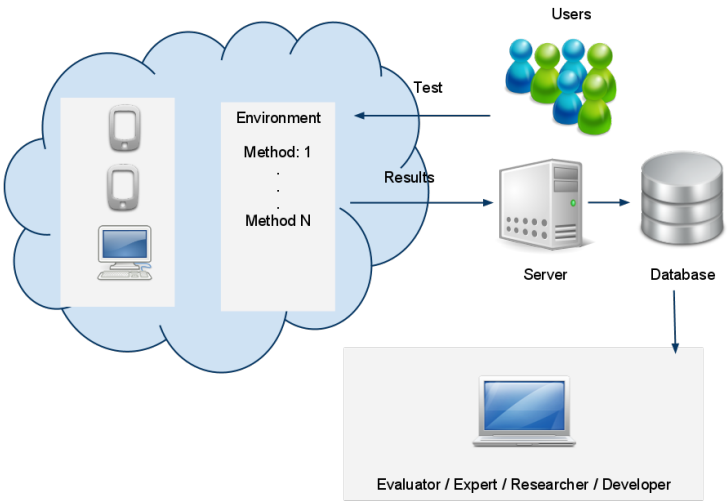



**Figure A.2:** The remote test approach uses a web server to facilitate the supervisor in the experiment not to be physically present in the test setting - but be present in real-time to guide the users.

### A.0.5 Remote Laboratory experiments

Logistically, there are several issues involving having supervisors being present at all times during a test. The method can be cumbersome, especially in a globalized world where developers and users are spread all over the globe. Therefore, a framework using remote observation of users by a supervisor can be undertaken, making laboratory experiments a smaller logistical burden for a developer.

The approach in Figure 2.11 still requires a supervisor to be present during evaluation of all test subjects. It is as labor intensive as the classic laboratory experiments, however it is more cost-efficient due to its less logistic requirements. The observations of users are restricted to the limitations of the installed equipment on the remote test location. This impacts eventual qualitative observations made by the supervisor.

### A.0.6 Asynchronous laboratory experiments

The use of passive observation of users is proposed to leverage the labour intensive laboratory tests, regardless whether they are remote or not, based on a web server that stores the results in a database as seen in Figure 2.12. When the test subject has conducted the test, the results are sent to the server and stored for later review by the supervisor/researcher.

The asynchronous method is not dependent on a supervisor being present during the test procedure, which removes some of the costs involved in evaluation of a given application. The tests conducted using this approach is a lot less extensive than the laboratory

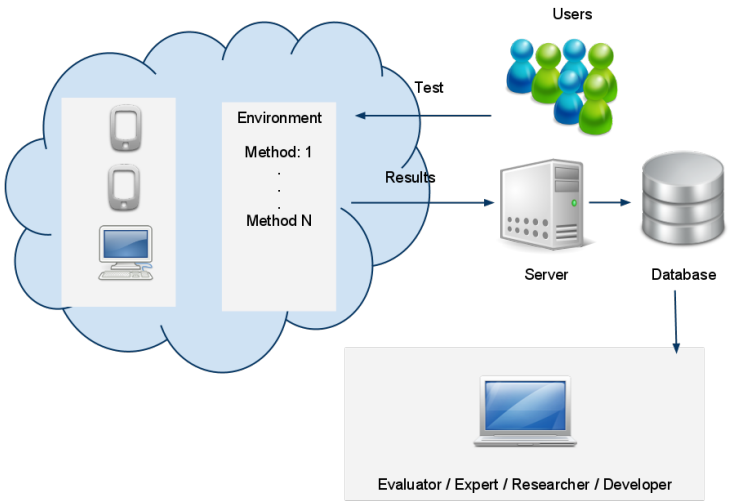



**Figure A.3:** The remote test approach uses a web server to facilitate the supervisor in the experiment not to be physically present in the test setting - but be present in real-time to guide the users through the experiment.

tests, that has been the evaluation method of choice in application development for decades. The recent focus on cost reduction of such practices, promotes remotely testing of users.

> *It is often highly relevant to get a cheap usability test although it is not complete. In that case, one of the remote tests would be an interesting possibility.* [Bruun *et al.*, 2009, p. 1627].

The tasks and test procedure can be simplified to become an asynchronous calibration routine that can eliminate the worst issues in an application in a cost efficient manner. Defining a generic *User model* to encompass the user requirements and getting information about the application in early development stages, can also be used to calibrate the system to tailor the user's needs and requirements.

The general overview of cost-efficient user testing approaches should prove relevant, when proposing a solution to create a cost efficient method of evaluation of critical errors in the prototype. In relation to the calibration method required from the problem statement the asynchronous client-side evaluation, there are advantages in letting the system evaluate itself against the user using a calibration routine. In order to see how such a method can be implemented the Adaptive Hypermedia System design will be described. The design considerations will then be adapted to the high fidelity prototype and client-side evaluation methods described & proposed in this section.

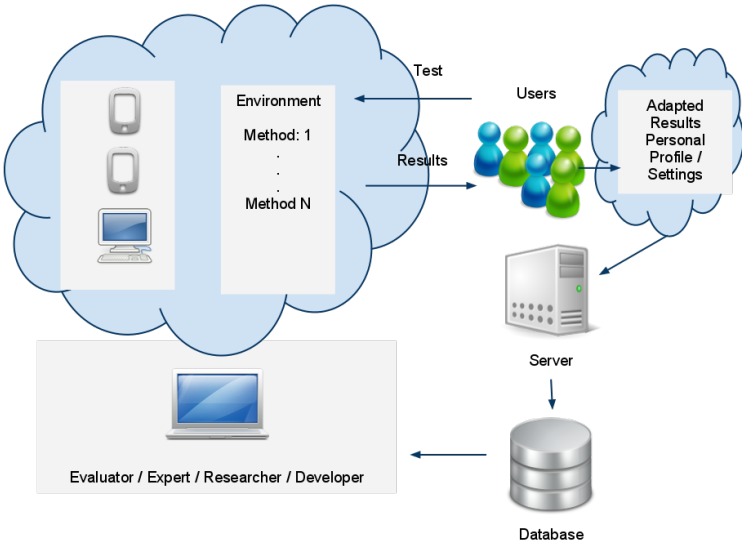



**Figure A.4:** The adaptable test results adds an abstract evaluation layer, client side, to the application that can be updated during runtime of the application.

# B. Development model

## B.1 Development Model

The purpose of designing the prototype for this project, is to measure if there are any differences in the interaction activity of solving a set of simple tasks among a group of users. In order to conduct these measurements, it is necessary for the different users to be able to solve the tasks without any previous training and considerable instructions. Therefore, the design process must ensure focus on the user, which can be done by Human-centered design [Sharp *et al.*, 2007, p. 462]. Human-centered design focuses on involving the users in the design of an affordance, application, etc, when designing with requirements centered on the needs and demands of the user. This involves asking a representative part of the segment for their needs and demands, which the design should address accordingly. Designing a novel product, where the context of use can be unclear to the user (and even the design team), creates demands for detecting the requirements without having to ask directly or specifically, since the user simply might not know what they want or are capable of - which is a classical design & evaluation problem.

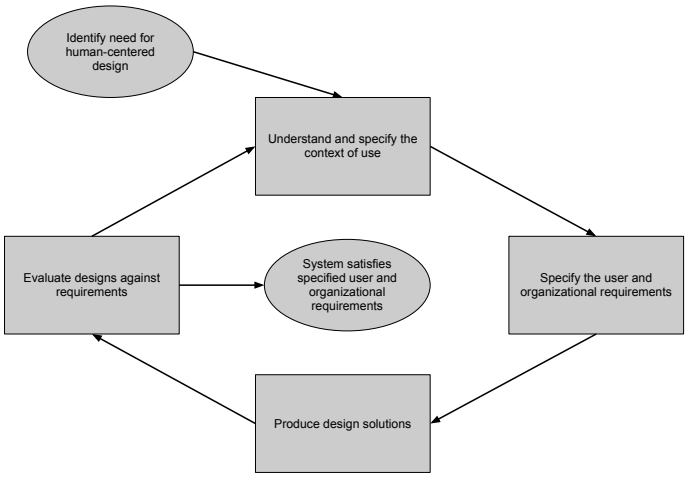



**Figure B.1:** The ISO 13407 procedure for Human-centered design are based on understanding the user context by specifying the user requirements, a production of a design solution and subsequent evaluation of the design solution. If the requirements are met the product finishes its development cycle otherwise it is repeated. [Sharp *et al.*, 2007, p. 463]

A generic approach of addressing planning and management in soft- and hardware projects is proposed by ISO 13407, which stipulates guidance on human-centered design activities throughout the lifecycle of an interactive product from the initial research phase to detection of user needs, production, and finally evaluation of the proposed solution/application [Sharp *et al.*, 2007, p. 463]. While the standard addresses the overall planning,

it does not concern the actual design process, but covers the following principles:

- The active involvement of users and the interaction between them and the developers are correlated with the effectiveness of the design process
- The design process is iterative and each step must be repeated upon further results of analysis and evaluation
- The allocation of function must be made as a balance between the technical capabilities of the system and human factors as reliability, flexibility and user well-being

While these statements may seem rather broad-spectered and nonspecific, it is the intend of the standard to introduce a formalized approach to the development procedure. The purpose of the standard is to devise the development into a manageable form so that planning of Human-centered design projects can be formalized and planned, prior to execution. This also makes it possible to make a cost-benefit analysis of the project in general. The production of a Human-centered design project according to the ISO 13407, is defined in 4 activities, central to the project:

1. Understanding and specification of the use context
2. Specification of user requirements
3. Production of design solutions
4. Evaluation of design against requirements

The activities are the foundation of a lifecycle model suggested by the standard, as visualized in Figure 2.8, which shows an iterative development cycle.

In the context of this specific project, where a generic calibration routine is sought to be verified, the user context will be defined by an application and the verification must be made using quantifiable data. This means that the results of the interaction can only be collected using a prototype that is capable of collecting such data. The environment should only reflect the need to verify that it is possible with such an approach and not be embedded in a specific problem field. The user requirements are not known, but is rather an area of interest for the experiment - when are the users finding a given task problematic to solve and is it beneficial to have the users tackle this problem or could it be generically avoided by a specific enough *User model*?

The generation a *User model* through a calibration routine is a very specific requirement that is fundamental for the proposed method to hold. The design solution will, in the first development cycle, only be a prototype in order to evaluate that a calibration method can be used to describe the users.

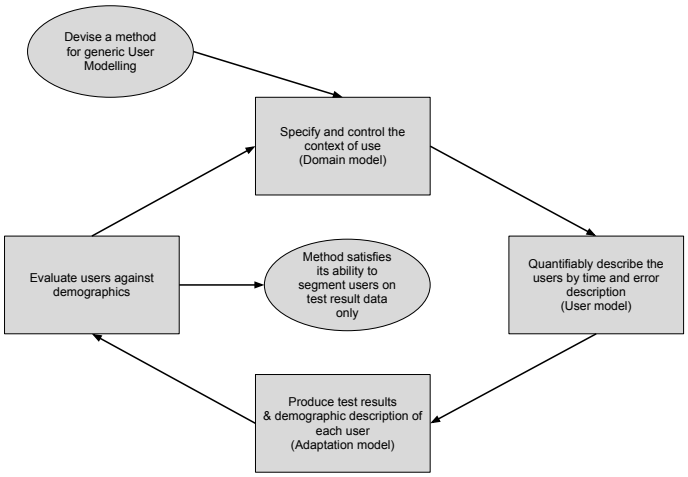



**Figure B.2:** The project seeks to device a generic method for *User model* calibration which is useful for any application. The method will be devised within the Adaptive Hypermedia System framework. The *Domain model* will be completely controlled to facilitate the test of the proposed method. The *User model* results dataset will only be composed of time measurement and error pr task. The test results along with demographic description parameters will be used to evaluate the method.

# C. DVD

# Bibliography

[Adler, 2011] Adler, Carlye (2011). *Ideas Are Overrated: Startup Guru Eric Ries' Radical New Theory*. Wired Magazine, http://www.wired.com/magazine/2011/08/st_qareis/, september 2011 edition.

[Allen, 2009] Allen, Matthew (2009). *Tim O'Reilly and Web 2.0: the economics of memetic liberty and control*. , Department of Internet Studies, Curtin University of Technology, http://netcrit.net/content/cpctimweb202009.pdf.

[Andersen, 2011] Andersen, Tania (2011). *Sådan programmeres det semantiske web*. Prosa, http://www.prosa.dk/aktuelt/nyhed/artikel/saadan-programmeres-det-semantiske-web/.

[Baranovskiy, 2011] Baranovskiy, Dmitry (2011). *Raphaël—JavaScript Library*. Sencha Labs, http://raphaeljs.com/.

[Berners-Lee & Fischetti, 1999] Berners-Lee, T. & Fischetti, M. (1999). *Weaving the Web: the original design and ultimate destiny of the World Wide Web by its inventor*. HarperSanFrancisco. `http://books.google.dk/books?id=oj--QgAACAAJ`.

[Brusilovsky, 1996] Brusilovsky, Peter (1996). Adaptive hypermedia: An attempt to analyze and generalize. In Brusilovsky, Peter, Kommers, Piet, & Streitz, Norbert, editors, *Multimedia, Hypermedia, and Virtual Reality Models, Systems, and Applications*, volume 1077 of *Lecture Notes in Computer Science*, p. 288–304. Springer Berlin / Heidelberg. `http://dx.doi.org/10.1007/3-540-61282-3_29`.

[Bruun *et al.*, 2009] Bruun, Anders, Gull, Peter, Hofmeister, Lene, & Stage, Jan (2009). Let your users do the testing: a comparison of three remote asynchronous usability testing methods. In *Proceedings of the 27th international conference on Human factors in computing systems*, CHI '09, p. 1619–1628, New York, NY, USA. ACM. `http://doi.acm.org/10.1145/1518701.1518948`.

[David & de Castro, 2009] David, Marcio Frayze & de Castro, Leandro Nunes (2009). A new clustering boids algorithm for data mining. In *Proc. of the 2009 AI*IA Workshop*

*on Complexity, Evolution and Emergent Intelligence*, p. 1–10. Published on CD, isbn 978-88-903581-1-1.

[Dumas & Redish, 1999] Dumas, J.S. & Redish, J. (1999). *A practical guide to usability testing*. Lives of Great Explorers Series. Intellect Books. http://books.google.com/books?id=4lge5k_F9EwC.

[Gaura & Newman, 2003] Gaura, Elena I. & Newman, Robert M. (2003). Using ai techniques to aid hypermedia design. In *Proceedings of the 21st annual international conference on Documentation*, SIGDOC '03, p. 100–104, New York, NY, USA. ACM. http://doi.acm.org/10.1145/944868.944890.

[Ghazarian & Noorhosseini, 2010] Ghazarian, Arin & Noorhosseini, S. (2010). Automatic detection of users' skill levels using high-frequency user interface events. *User Modeling and User-Adapted Interaction*, 20:109–146. 10.1007/s11257-010-9073-5. http://dx.doi.org/10.1007/s11257-010-9073-5.

[Hauger, 2011] Hauger, David (2011). *Using Asynchronous Client-Side User Monitoring to Enhance User Modeling in Adaptive E-Learning Systems*. Institute for Information Processing and Microprocessor Technology, Johannes Kepler University, Linz, Austria, http://www.fim.uni-linz.ac.at/Publications/Hauger/Hauger_UMAP09.pdf.

[Hauger *et al.*, 2011] Hauger, David, Paramythis, Alexandros, & Weibelzahl, Stephan (2011). Using browser interaction data to determine page reading behavior. In Konstan, Joseph, Conejo, Ricardo, Marzo, José, & Oliver, Nuria, editors, *User Modeling, Adaption and Personalization*, volume 6787 of *Lecture Notes in Computer Science*, p. 147–158. Springer Berlin / Heidelberg. 10.1007/978-3-642-22362-4_13. http://dx.doi.org/10.1007/978-3-642-22362-4_13.

[James & DebRoy, 2011] James, David A. & DebRoy, Saikat (2011). *RMySQL: R interface to the MySQL database*. R package version 0.8-0. http://CRAN.R-project.org/package=RMySQL.

[Lang, 2011] Lang, Duncan Temple (2011). *RJSONIO: Serialize R objects to JSON, JavaScript Object Notation*. R package version 0.95-0. http://CRAN.R-project.org/package=RJSONIO.

[LeBlanc, 2004] LeBlanc, D.C. (2004). *Statistics: concepts and applications for science*. Jones and Bartlett. http://books.google.com/books?id=gtawVU0oZFMC.

[Lehni & Puckey, 2011] Lehni, Jürg & Puckey, Jonathan (2011). *Paper.js — The Swiss Army Knife of Vector Graphics Scripting*. http://paperjs.org/.

[Leichtenstern & André, 2010] Leichtenstern, Karin & André, Elisabeth (2010). Mopedt: features and evaluation of a user-centred prototyping tool. In *Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems*, EICS '10, p. 93–102, New York, NY, USA. ACM. http://doi.acm.org/10.1145/1822018.1822033.

[Leichtenstern *et al.*, 2010] Leichtenstern, Karin, André, Elisabeth, & Rehm, Matthias (2010). Using the hybrid simulation for early user evaluations of pervasive interactions. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, NordiCHI '10, p. 315–324, New York, NY, USA. ACM. http://doi.acm.org/10.1145/1868914.1868952.

[Montero & Solana, 2007] Montero, Y. H. & Solana, V. H. (2007). *visualizious*. University of Granada. http://www.nosolousabilidad.com/hassan/visualizious/.

[mrdoob, 2011] mrdoob (2011). *three.js - Javascript 3D Engine*. https://github.com/mrdoob/three.js/#readmeQ.

[Norman, 2002] Norman, D.A. (2002). *The design of everyday things*. Basic Books. Basic Books. http://books.google.com/books?id=GeBmQgAACAAJ.

[Norman, 2005] Norman, D.A. (2005). *Emotional design: why we love (or hate) everyday things*. Basic Books. http://books.google.com/books?id=h_wAbnGlOC4C.

[Obitko, 2011a] Obitko, Marek (2011a). *Modularization of Ontologies*. Department of Cybernetics Faculty of Electrical Engineering Czech Technical University, http://www.obitko.com/tutorials/ontologies-semantic-web/modularization-of-ontologies.html.

[Obitko, 2011b] Obitko, Marek (2011b). *Semantic Web Architecture*. Department of Cybernetics Faculty of Electrical Engineering Czech Technical University, http://www.obitko.com/tutorials/ontologies-semantic-web/semantic-web-architecture.html.

[Ortiz Laguna *et al.*, 2011] Ortiz Laguna, Javier, Olaya, Angel, & Borrajo, Daniel (2011). A dynamic sliding window approach for activity recognition. In Konstan, Joseph, Conejo, Ricardo, Marzo, José, & Oliver, Nuria, editors, *User Modeling, Adaption and Personalization*, volume 6787 of *Lecture Notes in Computer Science*, p. 219–230. Springer Berlin / Heidelberg. 10.1007/978-3-642-22362-4_19. http://dx.doi.org/10.1007/978-3-642-22362-4_19.

[Papanikolaou *et al.*, 2000] Papanikolaou, K.A., Magoulas, G.D., & Grigoriadou, M. (2000). Computational intelligence in adaptive educational hypermedia. In *Neural Net-*

*works, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, volume 6, p. 629 –634 vol.6.

[Parker, 2007] Parker, Conrad (2007). *Boids Pseudocode.* http://www.kfish.org/, http://www.kfish.org/boids/pseudocode.html.

[Parra & Amatriain, 2011] Parra, Denis & Amatriain, Xavier (2011). Walk the talk. In Konstan, Joseph, Conejo, Ricardo, Marzo, José, & Oliver, Nuria, editors, *User Modeling, Adaption and Personalization*, volume 6787 of *Lecture Notes in Computer Science*, p. 255–268. Springer Berlin / Heidelberg. 10.1007/978-3-642-22362-4_22. http://dx.doi.org/10.1007/978-3-642-22362-4_22.

[R Development Core Team, 2011] R Development Core Team (2011). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0. http://www.R-project.org/.

[Resig *et al.*, 2011] Resig, John, Fry, Ben, Reas, Casey, Downe, Scott, Salga, Andor, Hodgin, Daniel, Kamermans, Mike, Humphrey, David, & Buckley, Jon (2011). *Processing.js a port of the Processing Visualization Language.* Processing community, http://processingjs.org/.

[Santos & Badre, 1994] Santos, Paulo J. & Badre, Albert N. (1994). Automatic chunk detection in human-computer interaction. In *Proceedings of the workshop on Advanced visual interfaces*, AVI '94, p. 69–77, New York, NY, USA. ACM. http://doi.acm.org/10.1145/192309.192327.

[Sharp *et al.*, 2007] Sharp, H., Rogers, Y., & Preece, J. (2007). *Interaction design: beyond human-computer interaction.* Wiley. http://books.google.com/books?id=kcEZAQAAIAAJ.

[Shaw *et al.*, 1977] Shaw, R., Bransford, J., & of Minnesota. Center for Research in Human Learning, University (1977). *Perceiving, acting, and knowing: toward an ecological psychology.* Lawrence Erlbaum Associates. http://books.google.com/books?id=YGN9AAAAMAAJ.

[Soegaard, 2010] Soegaard, Mads (2010). Affordances. This is an electronic document. Date of publication: March 22, 2010. Date retrieved: October 8, 2011. Date last modified: March 22, 2010. http://www.interaction-design.org/encyclopedia/affordances.html.

[W3C, 2006] W3C (2006). *Life Science Onotologies Linked via Semantic Web.* (MIT and ERCIM and Keio), http://www.w3.org/2006/Talks/1023-sb-W3CTechSemWeb/Overview.html#(32).

[W3C, 2011] W3C (2011). *W3C Working Draft HTML5 - the canvas element.* W3C, http://www.w3.org/TR/html5/the-canvas-element.html.

[Wickham & Hadley, 2007] Wickham & Hadley (2007). Reshaping data with the reshape package. *Journal of Statistical Software*, 21(12). `http://www.jstatsoft.org/v21/i12/paper`.

[Wickham, 2009] Wickham, Hadley (2009). *ggplot2: elegant graphics for data analysis.* Springer New York. `http://had.co.nz/ggplot2/book`.