ESTIMATION OF THE USER'S COGNITIVE LOAD WHILE INTERACTING WITH THE INTERFACE BASED ON BAYESIAN NETWORK

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS OF
THE MIDDLE EAST TECHNICAL UNIVERSITY
BY

AYSUN SAYDAM

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF COGNITIVE SCIENCE

SEPTEMBER 2021

ESTIMATION OF THE USER'S COGNITIVE LOAD WHILE INTERACTING WITH THE INTERFACE BASED ON BAYESIAN NETWORK

Submitted by AYSUN SAYDAM in partial fulfillment of the requirements for the degree of Master of Science in Cognitive Science Department, Middle East Technical University by,

Date: *10/09/2021*

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Aysun Saydam

Signature : _____

ABSTRACT

ESTIMATION OF THE USER'S COGNITIVE LOAD WHILE INTERACTING WITH THE INTERFACE BASED ON BAYESIAN NETWORK

Saydam, Aysun

MSc., Department of Cognitive Sciences

Supervisor: Assoc. Prof. Dr. Barbaros Yet

September 2021, 66 pages

The complexity of human machine interfaces is increasing significantly in parallel with the development of technology and excessive data growth, but human cognitive capacity is limited. Therefore, measuring cognitive load is one of the most preferential and common ways to test the usability of user interfaces. There are many different physiological, behavioral and subjective methods to measure human performance and workload. Moreover, there are cognitive predictive models and many related applications based on these models to predict performance and human workload on computer based tasks. The purpose of this study is to estimate the cognitive load and performance of the person by evaluating multiple methods together based on Bayesian network. For this, we modeled a Bayesian network that both uses a cognitive predictive model, and learns and regulates it with subjective data collected from people. After modelling, we conducted experiments with the interfaces of two different defense projects to collect data. We used the adapted Bedford scale at the end of each task of an interface and the NASA TLX rating scale for the overall rating of the interface after all tasks were completed. We confirmed that the Bayesian network effectively estimated the user's workload and performance. Our findings reveal that this model performs cognitive load analyzes much more efficiently in a short time. This study also demonstrates the differences between tasks and users, providing the opportunity to detect the complexity of subtasks and perform personalized performance and cognitive load analysis for each user.

Keywords: User Interface, Bayesian Network, Cognitive Load, Performance

ÖZ

ARAYÜZLE ETKİLEŞİME GİREN KULLANICININ BİLİŞSEL YÜKÜNÜN BAYES AĞINA DAYALI TAHMİNİ

Saydam, Aysun

Yüksek Lisans, Bilişsel Bilimler Bölümü

Tez Yöneticisi: Doç. Dr. Barbaros Yet

Eylül 2021, 66 sayfa

İnsan makine arayüzlerinin karmaşıklığı, teknolojinin gelişmesine ve aşırı veri büyümesine paralel olarak önemli ölçüde artmaktadır, ancak insanın bilişsel kapasitesi sınırlıdır. Bu nedenle, bilişsel yükü ölçmek, kullanıcı arayüzlerinin kullanılabilirliğini test etmenin en tercih edilen ve yaygın yollarından biridir. İnsan performansını ve iş yükünü ölçmek için birçok farklı fizyolojik, davranışsal ve öznel yöntem bulunmaktadır. Ayrıca, bilgisayar tabanlı görevlerde performansı ve insan iş yükünü tahmin etmek için bilişsel öngörü modelleri ve bu modellere dayalı çok çeşitli uygulamalar vardır. Bu çalışmanın amacı, Bayes ağına dayalı olarak birden fazla yöntemi bir arada değerlendirerek kişinin bilişsel yükünü ve performansını tahmin etmektir. Bunun için hem bilişsel bir tahmin modeli kullanan hem de bunu insanlardan toplanan öznel verilerle öğrenen ve düzenleyen bir Bayes ağı modelledik. Modellemenin ardından veri toplamak için iki farklı savunma projesinin arayüzleri ile deneyler gerçekleştirdik. Bir arayüzün her görevinin sonunda uyarlanmış Bedford ölçeğini ve tüm görevler tamamlandıktan sonra arayüzün genel derecelendirmesi için NASA TLX derecelendirme ölçeğini kullandık. Bayes ağının kullanıcının iş yükünü ve performansını etkili bir şekilde tahmin ettiğini doğruladık. Bulgularımız, bu modelin bilişsel yük analizlerini kısa sürede çok daha verimli bir şekilde gerçekleştirdiğini ortaya koymaktadır. Bu çalışma ayrıca görevler ve kullanıcılar arasındaki farkları göstererek, alt görevlerin karmaşıklığını tespit etme ve her kullanıcı için kişiselleştirilmiş performans ve bilişsel yük analizi gerçekleştirme fırsatı sunar.

Anahtar Sözcükler: Kullanıcı Arayüzü, Bayes Ağı, Bilişsel Yük, Performans

To My Late Friend Mesut Özgür Sevim

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my sincere gratitude to many precious people in my life who supported me while writing this thesis. First of all, I would like to thank my supervisor, Assoc. Prof. Dr. Barbaros Yet for his support and guidance during this process. I would not have achieved my aim without him, his deep knowledge and guidance helped me to write my thesis.

I would also like to thank entire Cognitive Science Department for this wonderful learning adventure and my committee members Assist. Prof. Dr. Murat Perit Çakır and Assoc. Prof. Dr. Ayça Kolukısa Tarhan for valuable comments and contributions.

I also want to express my gratitude to my manager, Özgür Ülvan who supported me in writing this thesis with his creative ideas. In addition, I want to thank my colleagues, who are also my dear friends, at ASELSAN. Especially, I would like to thank Gülce for her valuable support during this difficult process. Moreover, I am grateful to my friend, Mesut Özgür for conducing me to start this master's program and Şeyma for always being with me throughout this journey.

My deepest thanks are for my beautiful mother who supports me in all phases of my life unconditionally and Olaf for not leaving me alone for a moment while writing the thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BN	Bayesian Network
BCMS	Behavior Cognitive Model Scale
CMN-GOMS	Card, Moran, Newell GOMS
CPM-GOMS	Cognitive - Perceptual - Motor GOMS
DAG	Directed Acyclic Graph
ECG	Electrocardiogram
EEG	Electroencephalography
GOMS	Goals, Operators, Methods, Selection rules
HCI	Human Computer Interaction
KLM	Keystroke Level Model
MCMC	Markov Chain Monte Carlo
NASA	National Aeronautics and Space Administration
NASA - TLX	NASA Task Load Index
NGOMSL	Natural GOMS Language
SWAT	Subjective Workload Assessment Technique
VACP	Visual, Auditory, Cognitive, Psychomotor

CHAPTER 1

INTRODUCTION

1.1. Motivation of the Study

Cognitive limitations of users are one of the most significant parameters for designing better user interfaces (Akgun, Akilli & Cagiltay, 2011). Especially information requirements must be considered by centralizing cognitive limitations and capabilities of end users during the design process (Patel & Kushniruk, 1998). Therefore, understanding the workload and effort requirement while designing an interface is vital for the effective development of the interface.

There are numerous physiological, behavioral, predictive and subjective methods to measure and estimate cognitive load and performance such as observing heart rate, monitoring brain activity, eye tracking, and mouse tracking, or modeling human cognition. Physiological methods provide objective and reliable measurements but they are also costly and require special measurement equipment. Subjective measurement instruments such as questionnaires are more cost effective, but they require considerable human effort and can be time consuming. Time for executing different tasks can also be used as additional indirect information about cognitive load. Furthermore, there are cognitive predictive models to estimate time and workload. These models only make estimation for an average expert user. They do not adjust and make inferences for possible differences between the users. In addition, reflecting the dynamic variables and visual complexity of the user interface is not easy in these models. These variables and overall complexity are ignored if they are not directly related to the modeled task scenario. However, they are factors that seriously affect cognitive load. This study focuses on developing an approach to combine a variety of these sources of information for cognitive load and performance estimation.

1.2. Purpose of the Thesis

In this thesis, we aim to aid improving the usability of complex user interfaces by providing an accurate estimate of cognitive load and performance from limited amount of data. We focus on obtaining a time and workload estimate through a cognitive model and revise this estimate with the completion and questionnaire data we collected from users to make more accurate estimations. We use Bayesian Network (BN) technology for this purpose as BNs offer suitable framework for synthesizing different sources of information. The thesis proposes a systematic approach to build Bayesian

network that updates the estimates from available cognitive models such as GOMS with the data about completion times and subjective instruments, and illustrates the use of this approach with a case study.

1.3. Contributions

The main contribution of this study is a systematic and novel approach that combines data from cognitive models, subjective instruments and interface use to estimate workload and performance using Bayesian Networks. Measurement successes of traditional methods can vary according to the many parameters such as user's characteristics, nature of the interface, the complexity of the tasks, and the different attributes of the design. There are already several studies based on mixed methods; in which researchers combine multiple methods for more consistent and reliable measurement and estimation. This study uses Bayesian Networks, which are particularly suitable for combining multiple sources of data based on probabilistic inference. The proposed approach also takes the differences between users and tasks into account when making performance and cognitive load estimates.

Previous approaches estimate cognitive load and effort either by model-based predictive tools or by collecting physiological, behavioral or subjective data from the user with different methods, as discussed in more detail in Chapter 2. The proposed Bayesian model provides a suitable approach to synthesize multiple sources of cognitive load data by reflecting the differences between users, interfaces and tasks, and it allows personalized predictions for cognitive load and performance.

In particular, the proposed method combines information from GOMS model, Bedford subjective instrument, and task completion time data collected from users. Our Bayesian model starts to estimate with GOMS based cognitive model, then reviews data with Bedford data collected from the users. GOMS is a modelling approach that includes a set of methods which have different abilities such as CMN-GOMS, NGOMSL, and CPM-GOMS (Kieras, 1999). It can be used as a predictive model for computer based tasks. We use a GOMS based cognitive tool, called Cogulator, to derive the prior values for our model. It is an open source program that predicts execution time, working memory load and mental workload. We use Bedford scale to collect observed data for our model. It is a unidimensional psychometric scale which is suitable to verify workload of computer-based tasks and gives workload measurement result from 1 to 10. We also examine the use of NASA TLX rating scale which is another subjective scale to measure overall workload. It is a multi-dimensional scale which calculates overall workload with six indicators according to their weights and gives result from 1 to 100.

The second contribution of this thesis is estimating the workload and completion times of multiple tasks in two interfaces used in the defense industry, and evaluating the performance of the proposed approach. We used Cogulator to derive prior data for five different subtasks of two defense user interfaces with varying degrees of complexity.

Then we collected adapted Bedford scale data for five different tasks of each user interface and NASA-TLX rating scale data for the general evaluation of each user interface from 20 participants. We also recorded the execution time to complete tasks of users. We used linear regression to analyze the relation between overall workload of the interface measured by NASA-TLX and the workload of tasks measured by the Bedford scale. By analyzing the Bedford data and completion time with the Bayesian model, we obtained posterior estimation results of cognitive load and performance and compared it with the predictions provided by the GOMS model. In addition, we also analyzed the differences between tasks and users.

1.4. Outline

In the remainder of this thesis, the second chapter presents an overview of Bayesian models and BNs and describes the use of Bayesian models for user interfaces. The second chapter also reviews the cognitive models and subjective instruments available for cognitive load and effort estimation. The third chapter presents the proposed methodology for building BN models for cognitive load estimation, and describes the case studies, and the evaluation approach used in these case studies. The fourth chapter presents the results of the case studies. Finally, the fifth chapter presents our conclusions and discusses potential future studies.

CHAPTER 2

LITERATURE REVIEW

According to the computational theory of mind; the mind corresponds to a computer, mental representations correspond to computer programs and thinking is specified as a computational process. Mental representations can be considered as generative models which can support inferences in diverse situations according to the generative approach of cognition. These generative models are uncertain with many possible outcomes as the values of the large part of the variables in these models are unobserved. Probability is a suitable tool to represent this uncertainty. Moreover, the outcomes of these models can be updated once, we acquire further information about those variables, which corresponds to Bayesian inference. Consequently, Bayesian models offer a suitable approach to model many aspect of cognition such as learning and reasoning under uncertainty (Goodman & Tenenbaum, 2016).

This study focuses on the use of Bayesian models to update the model and review the uncertainty regarding cognitive workload. This section gives a recap of Bayesian networks (Section 2.1), and reviews the approaches used for estimating cognitive load and effort in HCI including the previous use of Bayesian models in this domain (Section 2.2).

2.1. Bayesian Models

Bayesian inference is a prevalent and practical method for data analysis in many scientific fields (Lee & Wagenmakers, 2014). Briefly, Bayesian inference is updating the probability distributions of unobserved variables based on a probabilistic model of the variables, and observations made on a part of its variables (Gelman, Carlin, Stern & Rubin, 1995). Bayesian models are suitable for combining different sources of information and reflecting relations between variables explicitly. Bayesian models also offer flexible data collection; researchers can continue or terminate data collection according to the confidence of their posteriors, and they can terminate when the evidence is satisfactory enough (Lee & Wagenmakers, 2014).

We describe the main principles of Bayesian approach in the following sections by focusing on the methods and distributions we used in this study.

2.1.1. Bayes' Theorem

Bayesian data analysis is a resilient process to make inferences from data using probabilistic models for quantities which are unknown or observed. There are two main principles of Bayesian analysis; first one is uncertainty in other words "degree of belief", is measured by probability, and the second one is prior belief is updated by using observed data to get posterior data (Lee & Wagenmakers, 2014).

For instance, we want to see Bayesian statistical conclusions about a parameter θ according to D which means the observed data. First of all, our prior belief about θ must be expressed as a probability distribution which is specified as $p(\theta)$. Second, our updated belief according to the observed data is the posterior distribution and we symbolize it as $p(\theta \mid D)$. Moreover, $p(D \mid \theta)$ indicates likelihood, $p(D)$ denotes marginal likelihood which is also called evidence (Lee & Wagenmakers, 2014). Formulization of posterior distribution based on these information, named Bayes' Theorem is given below.

$$p(\theta \mid D) = p(D \mid \theta) \times p(\theta) / p(D)$$

In other words;

posterior = (**likelihood** x **prior**) / **marginal likelihood**

Gelman and colleagues (1995) classify Bayesian data analysis process as three phases; the first one is constructing a complete probabilistic model, second one is updating and conditioning on the observed data, and the last one is evaluating the result and consistency between data and model. In this aspect, we begin with a model that provides a joint probability distribution for both θ and D to make inference about θ given D and we reflect the joint probability density function as a product of two densities which are the prior distribution $p(\theta)$ and data distribution $p(D \mid \theta)$ to calculate posterior distribution $p(\theta \mid D)$ by conditioning on observed data $p(D)$.

In conclusion, as Lee and Wagenmakers put it; "Bayes' rule provides a bridge between the unobserved parameters of models and the observed data." (2014, p. 45). However, in cases where there are many variables and high complexity, computation of Bayes' theorem and representation of models becomes challenging. In this case, DAGs can assist representation and computation issues.

2.1.2. Directed Acyclich Graph (DAG)

Graphical models in the form of Directed Acyclic Graphs are suitable for representing large and complicated Bayesian models with a set of nodes and a set of edges that respectively denote variables, and the probabilistic relations between them. A graph is called directed if there are only directed edges and acyclic if there is no cycle. If there is a directed edge from i to j, but no edge from j to i, i is named parent of j. There can

be conditional dependence or independence between variables. If there is no edge between variables they are conditionally independent of each other. Figure 1 shows simple DAGs over three parameters based on different dependencies.

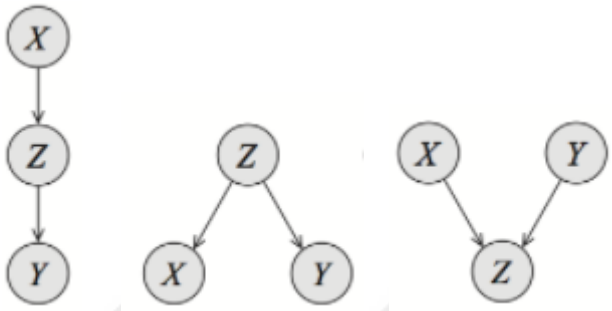

Figure 1: DAGs

Let's define a DAG as $D = (V, E)$, while $V = \{1,\ldots,q\}$ is a set of nodes , E means a set of directed edges and $E \subseteq V \times V$.

Let $U = (U_q)$, $q \in V$ be a set of random variables. U is a BN with respect to D where $pa(j)$ is the set of parents of j.

$$\mathbf{p}(\boldsymbol{U}) = \prod_{j \in V} \mathbf{p}\,(\boldsymbol{U_j} \mid \boldsymbol{U_{pa(j)}})$$

Chain rule helps to calculate the probability of any member of a joint distribution from conditional dependencies for any set of random variables.

$$\mathbf{P}\,(\boldsymbol{U_1} = \boldsymbol{u_1}, \ldots, \boldsymbol{U_q} = \boldsymbol{u_q}) = \prod_{j=1}^{q} \mathbf{P}\,(\boldsymbol{U_j} = \boldsymbol{u_j} \mid \boldsymbol{U_{j+1}} = \boldsymbol{u_{j+1}}, \ldots, \boldsymbol{U_q} = \boldsymbol{u_q})$$

This can be written as above;

$$\mathbf{P}\,(\boldsymbol{U_1} = \boldsymbol{u_1}, \ldots, \boldsymbol{U_q} = \boldsymbol{u_q}) = \prod_{j=1}^{q} \mathbf{P}\,(\boldsymbol{U_j} = \boldsymbol{u_j} \mid \boldsymbol{U_k} = \boldsymbol{u_k} \text{ for each } \boldsymbol{U_k} \text{ which is a parent of } \boldsymbol{U_j})$$

This is how BN represents the probability distribution according to the DAG and this is the factorization definition of BN.

2.1.3. Bayesian Networks

Bayesian networks are graphical models that represent condensed joint probability distributions over the set of variables by considering the conditional dependencies between them, via a DAG (Pearl, 1988). Namely, they are based on probability theory and graph theory in combine and they are used for many tasks such as prediction, reasoning, diagnostics, anomaly detection, automated insight and decision making under uncertainty (Stephenson, 2000). Studies in this area are becoming increasingly popular and significant progress has been made, especially in the last 50 years. BNs are useful to combine different sources of information and handle missing part of the information (Lee & Wagenmakers, 2014). In addition, they provide an explicit representation of uncertain information and express this uncertainty via model outputs.

BN development is an iterative process. Modular architecture of BN facilitates this iterative development process (Chen & Pollino, 2012). We can build more than one model which are different in many dimensions to construct the most useful one. Even, it is possible to split BN into subnetworks which represent diverse system components (Chen & Pollino, 2012). If our data is partial or uncertain, we can still use Bayesian network for many cases. Data-driven learning algorithms are also available to learn BNs from data. If we can decide correctly which model, method and prior parameter distribution to use, we will get more appropriate results with data.

The focus point of this study is Bayesian data analysis which help us to make sense of data. It is a process which conforms a probabilistic model into a dataset and reflects the condensed result via a probability distribution based on the parameters, predictions, and observations (Gelman, Carlin, Stern & Rubin, 1995). For this purpose, parameters in Bayesian data analysis models can be defined as latent variables of interest and we infer them through observed data. We have prior and posterior parameters and two distributions of both parameters to examine. Prior parameter distribution is our initial belief about parameters, posterior parameter distribution is our updated belief after observations. Similarly, prior predictive distribution means what data to expect based on our initial beliefs before observing any data and posterior predictive distribution means what data to expect given the observed data (Goodman & Tenenbaum, 2016).

BN Development

Causality and conditioning are the key terms to develop BN. Knowledge is encoded as causal models in probabilistic programming which is practical to see causal relations. Causal relations are directed, because X causes to Y is not equivalent to Y causes X, they are completely different. Meanly, while data can flow both directions, the causal effect can have one direction. In the light of this information, BN development starts with creating a DAG which reflects causal relations, conditional dependence and independence between nodes. But, at first

we need to synthesize our existing knowledge according to the scope and purpose of our model.

We start by defining conditional probabilities of each node and the states. The relations between nodes are defined on conditional probability tables attached to nodes which specify probability or "degree of belief". We have to specify prior distribution for each parameter in BN. We use Gaussian distribution and Gamma distribution in this study. Gaussian distribution is suitable for measurements whose mean and standard deviation are known only and Gamma distribution is a suitable prior distribution for standard deviation in general (see e.g. Chapter 2 of Lee and Wagenmakers, 2014 for a detailed description of Gaussians models). Prior specification of model parameters is challenging. These priors can be based on subjective approach like past experience or knowledge. But objective approach is suggested for priors for more consistent and reliable model (Chen & Pollino, 2012). Then, we enter observations or other evidences into Bayesian network to get updated state of each node based on Bayes' theorem.

BN Inference

Once the priors of the parameters in a BN model is defined and the data about the observed variables is instantiated, the posterior distribution of parameters can be computed by inference algorithms such as rejection sampling, MCMC (Markov Chain Monte Carlo), variational inference, Metropolis Hastings or Hamiltonian Monte Carlo (Goodman & Tenenbaum, 2016). Each of these algorithms have certain advantages and disadvantages for different types of models (see Chapter 8 of Goodman & Tenenbaum, 2016 for a review of inference algorithms). For instance, sampling algorithms like Gibbs sampling or importance sampling make compute posteriors approximately. Exact algorithms such as Junction Tree computes exact posteriors by transforming the BN into a tree structure and making factor operations on it. Inference algorithms can also be used to compute the posterior distributions of unobserved variables once some of the variables are observed. We use MCMC inference algorithm in this study which is a family of general purpose sampling algorithms based on a Markov chain whose stationary distribution is aimed to be the posterior distribution (see e.g. Chapter 6 of Goodman and Stuhlmüller, 2014 for a detailed description of MCMC).

BNs also learn and regulate data according to the observations. Learning can be considered as conditional inference in a model which has hypothesis, fixed latent variable and set of observations (Goodman & Tenenbaum, 2016). After we enter training data into BN, we can get answers based on the hypothesis.

2.2. Interface, Workload and Performance

In this section, cognitive models, psychometric instruments used to measure workload, physiological measurements, mixed studies combining them and HCI studies based on Bayesian approach are reviewed.

2.2.1. HCI and Usability

Effective design of human computer interaction is one of the main challenges of user interfaces and there are many aspects of user interface design based on human computer interaction within cognitive science perspective (Patel & Kushniruk, 1998). Fisher stresses the importance of human computer collaboration which means two or more agents' common work to fill a need of achieving shared desired goals (2001). There are two viewpoints within human computer collaboration; the first approach is emulation and the second is complementing; while the emulation approach aims to design computers like humans, the complementing approach accepts that computers aren't humans and human centered design is the key point to improve collaboration and interaction with creative design (Fisher, 2001). Historically, the emulation approach was the focus point, but limited achievements of the emulation approach led to complementing approach to become more popular and desirable. The focus point of earlier HCI studies was the design criteria of graphical user interfaces which stresses the most usable choices of design items such as menus and icons, then design focus shifted beyond items of interface in time (Fischer, 2001). The focus point gradually shifted to usability, which considers the different aspects of the relationship between the system and the user.

A well-designed interface is expected to meet multiple usability criteria. The U.S. Military Standard for Human Engineering Design Criteria (1999) identifies usability goals with achieving desired performance for operation, maintenance, control and minimum skill requirement to learn and use (Shneiderman & Plaisant, 2010). In addition, user satisfaction and trust also determine the efficiency of interaction design (Gokcay & Yildirim, 2011).

One of the most important milestones within HCI is understanding and modelling human behavior and limitations which led to new discoveries and interaction techniques in time (MacKenzie, 2012). According to Riva and colleagues (2005); interaction management and multimodal input/output are the key terms of the usable and intelligent user interfaces. Interaction management means adaptive user interface which can be adapted to different situations by managing interruptions, errors and monitoring the user behavior to anticipate next action, warn the user or change probable consequences (Riva, Vatalaro & Davide, 2005). So, computers need to be adapted to people for more efficient and satisfactory experience. But there are a large number of users which have different abilities or disabilities and it is difficult to understand and address this diversity (Fischer, 2001). User modeling and analysis is a crucial element of HCI to understand and address this

problem (Fischer, 2001) as they enable better understanding the interaction between systems and their users. Moreover, Kieras and colleagues identify the empirical user testing as a standard method to create a usable system which is based on iterative testing and redesign process with actual users of the system (1995).

In summary, understanding the user is the focus point of usability. An important element of understanding the users is to measure and estimate their cognitive load and performance. There are multiple approaches for measuring and predicting workload including cognitive predictive models, physiological measurements, subjective measurements and mixed method studies. The remainder of this chapter reviews these studies.

2.2.2. Cognitive Models

Cognitive load is an important part of user interface analysis, so many cognitive scientists focused on human's ongoing cognitive tasks, their cognitive capacity, cognitive cost of the system and human's cognitive limitations (Gokcay & Yildirim, 2011). Cognitive models for human cognition have been developed for computer based tasks to simulate human behavior and performance (Yuan, Li & Rusconi, 2020). These models allow the analysis of cognitive load at early stages of design before implementation and user testing. This section examine the KLM and GOMS models, which are popular cognitive models in this domain.

KLM

KLM was presented in 1980 by Card, Moran and Newell and it only consists of keystroke level operators to model actions like mouse click, buttons and keystrokes based on serial stage model (Yuan, Li & Rusconi, 2020). There is no goals, methods or selection rules in KLM. There are only 6 operators; K is keystroke, P is pointing a target via mouse, H is homing the hands to keyboard or mouse, D is drawing, M is mental preparation for physical actions and R is response time of the system (Yuan, Li & Rusconi, 2020). Every single operator has a default estimation of execution time. To sum up, KLM estimates time for a particular task by listing sequence of primitive operators and summing the execution times of these operators (John & Kieras, 1996). Unfortunately, it is not usable to analyze abstract and complex tasks.

GOMS

After KLM, GOMS came into stage which is completely different. *The Psychology of Human-Computer Interaction* book written by Card, Moran and Newell in 1983 can be taken as a milestone in this field which presents the GOMS method (John & Kieras, 1996). GOMS is another theoretical method in HCI to analyze routine interaction processes in terms of Goals, Operators, Methods and Selection rules which was used in many studies for different purposes and formed the basis of many subsequent studies. It has become one of the most used and popular modeling techniques to analyze the complexity of user interfaces in time (John & Kieras, 1996).

GOMS consists of four principles; Goals can be defined as what the user is trying to achieve and they can be divided into subgoals, Operators are the basic cognitive, motor, or perceptual actions used to achieve goals such as Point, Click, Type, Methods can be considered as procedures which define how to achieve goals and Selection rules denote which method should be used to achieve a particular goal according to the context (Hochstein, 2002). Methods consist of Operators used by user for desired Goals based on a hierarchical structure, and if there is more than one method to achieve a goal, Selection Rules are used to choose appropriate method according to the situation (Kieras, 1999). GOMS model can make a prediction of time needed to achieve a particular goal, verify the functionality of design to achieve goals, possibility to perform tasks at a certain time and help designer or developer to prepare tutorials about system by representing the explicit user activity (Hochstein, 2002).

GOMS is not a single method, it is a family of modeling methods to analyze system complexity based on user's behavior (Hochstein, 2002). There are many variants of this family such as CMN-GOMS, NGOMSL, and CPM-GOMS (John & Kieras, 1996). CMN-GOMS (Card, Moran, Newell GOMS) is used to identify the original GOMS formulation which was developed by Card, Moran and Newell (1983). It is a more advanced method based on KLM that has subgoals and selection rules in addition (Hochstein, 2002). It can predict not only the execution time but also the operator sequence. NGOMSL (Natural GOMS Language) is a notion of natural language procedure based on CMN-GOMS to represent GOMS models which predicts execution time, operator sequence and time required to learn the methods (John & Kieras, 1996). CPM-GOMS (Cognitive - Perceptual - Motor GOMS) is also based on other GOMS models, but it does not evaluate operators just serially, it makes an extra assumption that cognitive, perceptual and motor operations can also be performed in parallel (John & Kieras, 1996).

Construction of GOMS models is relatively easy and these models are effective to use, but their limitations also exist (Kieras, 1999). A designer has to prepare task analysis to make clear which goals are desired to be achieved, because GOMS models start after task analysis, furthermore, GOMS can only predict procedural measurements of usability and GOMS analysis can be used on clearly defined tasks for only experienced users (Kieras, 1999). They cannot evaluate the user's knowledge level about system.

There are many applications based on different GOMS techniques such as Cogulator (Liaghati, Mazzuchi & Sarkani, 2020), CogTool (Kovesdi & Joe, 2019), GLEAN (Kieras, Wood, Abotel & Hornof, 1995), and SANLab-CM (Yuan, Li & Rusconi, 2020). Among these, CogTool is a widely used open source program based on KLM and its accuracy claim between predicted time and observed execution time is within 20 percent (+/-10%) (Kovesdi & Joe, 2019). CogTool is easy to use but it requires visual representation of interactions. Accordingly, it can be used after design completion. Moreover, it is not able to model parallel tasks and just provides primitives for computer based tasks (Kovesdi & Joe, 2019). Furthermore, Jorritsma and colleagues showed that CogTool and KLM methods are not reliable for analysis in

some cases (2015). They used KLM, GOMS and CogTool for three tasks of three different interfaces to predict user performance. Then, they conducted experiments with 20 people and showed that the predicted performance did not correspond to the actual performance in the majority of the tasks. There were no statistically significant difference between the predictions of different approaches In addition, CogTool does not estimate working memory load and mental workload, and it is not suitable to model cognitive tasks like memorizing and multitasking (Kovesdi & Joe, 2019).

Cogulator is another open source program based on GOMS that predicts execution time, working memory load and mental workload (Kovesdi & Joe, 2019). Cogulator enables users to build multiple GOMS models like KLM, NGOMSL, CMN-GOMS and CPM-GOMS. It offers a simple interface (see Figure 2 and 3 for the monitoring screen and activity interface of Cogulator) and it is capable of modeling multitasking and memorizing (Kovesdi & Joe, 2019). Default time estimates, creation of new operators and modification of parameters could be done through its interface without requiring to modify its source code. Due to these advantages, we use Cogulator for building cognitive models in this study.

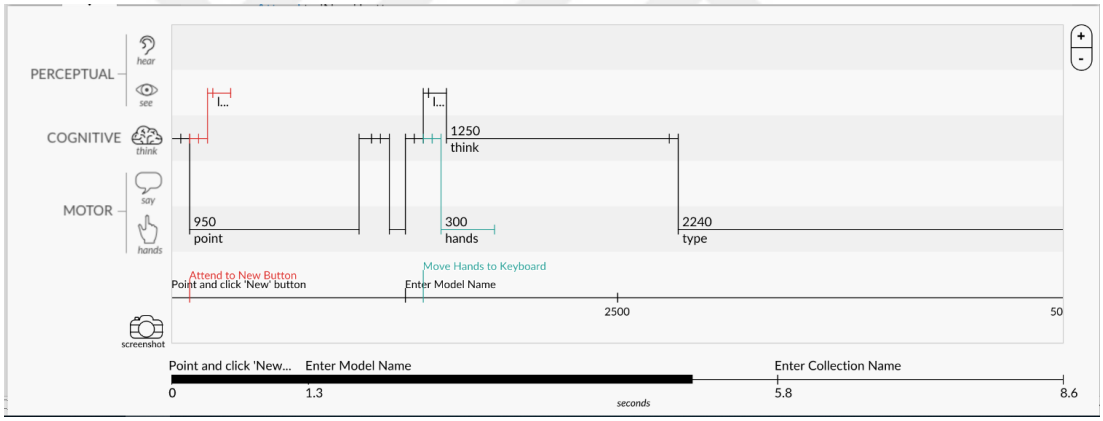

Figure 2: Screenshot of Cogulator's monitoring screen

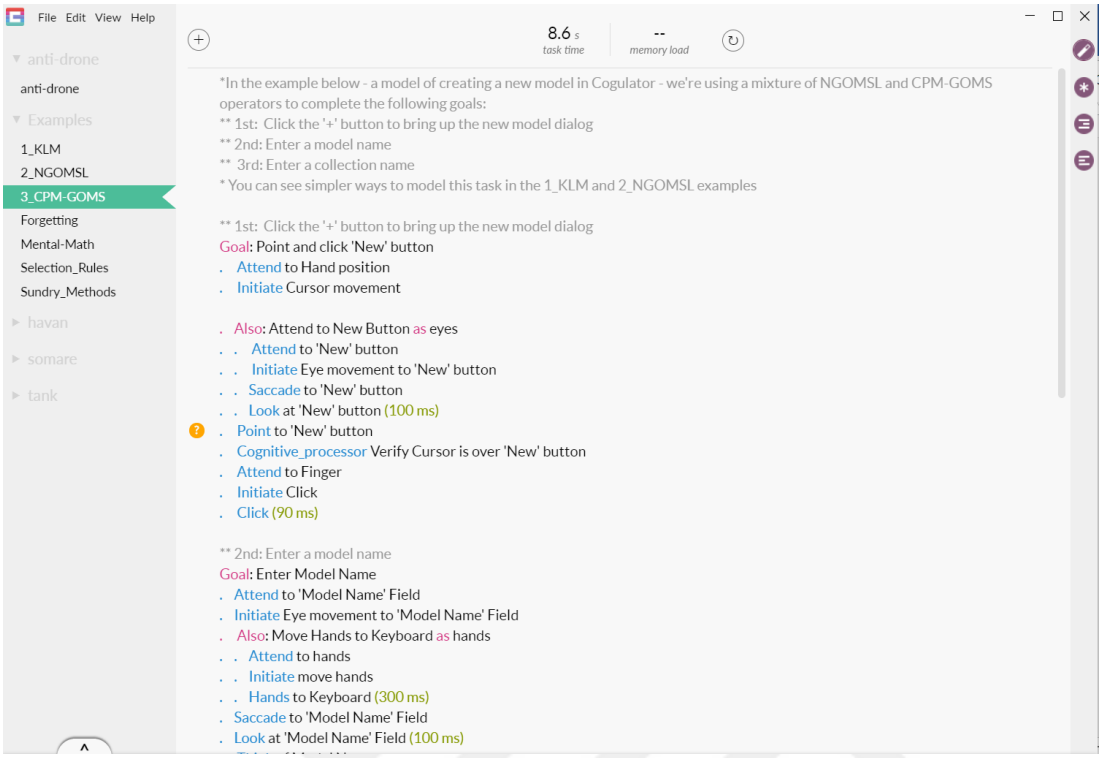

Figure 3: Screenshot of Cogulator's text-based interface

2.2.3. Subjective Workload Assessments

Subjective workload assessments are psychometric measurement instruments that reflect the user's opinion about workload. These instruments can be in the form of questionnaires composed visual, verbal or Likert rating scales. Popular subjective workload assessment instruments include Paas Scale (Sweller, 2018), NASA-TLX Rating Scale, Cooper-Harper Scale, and the Bedford Workload Scale (Moré, 2014).

Psychometric properties of these instruments are analyzed to quantify their reliability and validity. Reliability corresponds to the variation of the results between different use of the instruments by different users or by the same user. Validity corresponds to the accuracy and precision of the measurements of the instrument and the actual latent trait it aims to measure. Psychometric properties of subjective workload assessment instruments can be classified as sensitivity, diagnosticity, interference, equipment requirements and operator acceptance (Zhang et al., 2015). Sensitivity reflects the power of instrument to detect changes in demand or difficulty. Diagnosticity includes definition of changes and the reason of these changes. Interference is the degree of interfering with the primary task performance, which is the central object of assessment. Equipment requirements involve aspects like time, software, and instruments. Operator acceptance refers to the user's opinion of the usefulness of the method (Rubio, Díaz, Martín & Puente, 2004).

Although these subjective workload assessments are sometimes criticized for not being objective and being biased from person to person, they are frequently preferred as they are cost-effective, easy-to-apply and their results are found useful in many studies (Ramkumar et al., 2017). Furthermore, subjective data can especially be necessary and appropriate in certain cases. Because it is the only source to understand the personal views of people. This study uses adapted Bedford Workload Scale and NASA-TLX and the following sections describe the details of these subjective questionnaires.

Bedford Workload Scale

Bedford Workload Scale (Figure 4) is a unidimensional scale, modified from Cooper-Harper rating scale that measures if a task is possible to complete and workload is satisfactory or tolerable. Bedford Workload Scale is primarily developed for complex tasks which requires serious cognitive resources, high concentration and multitasking skills such as piloting activities (Miller, 2001; Zhang et al., 2015). NASA considers Bedford scale as the most appropriate assessment instrument during the verification phase after design (NASA, 2020).

Bedford Scale is composed of 10 questions each having scale ratings ranging from 1 to 10. The questions asked to a user are determined based on a hierarchical decision tree which identifies user's spare mental capacity while completing a task ("Cognitive Workload", 2020). Users navigate through the hierarchical tree and select a single rating based on the explanations by narrowing down their choices step by step. Bedford rating scale clarifies if the workload is satisfactory, tolerable, possible or impossible (Casner & Gore, 2010).

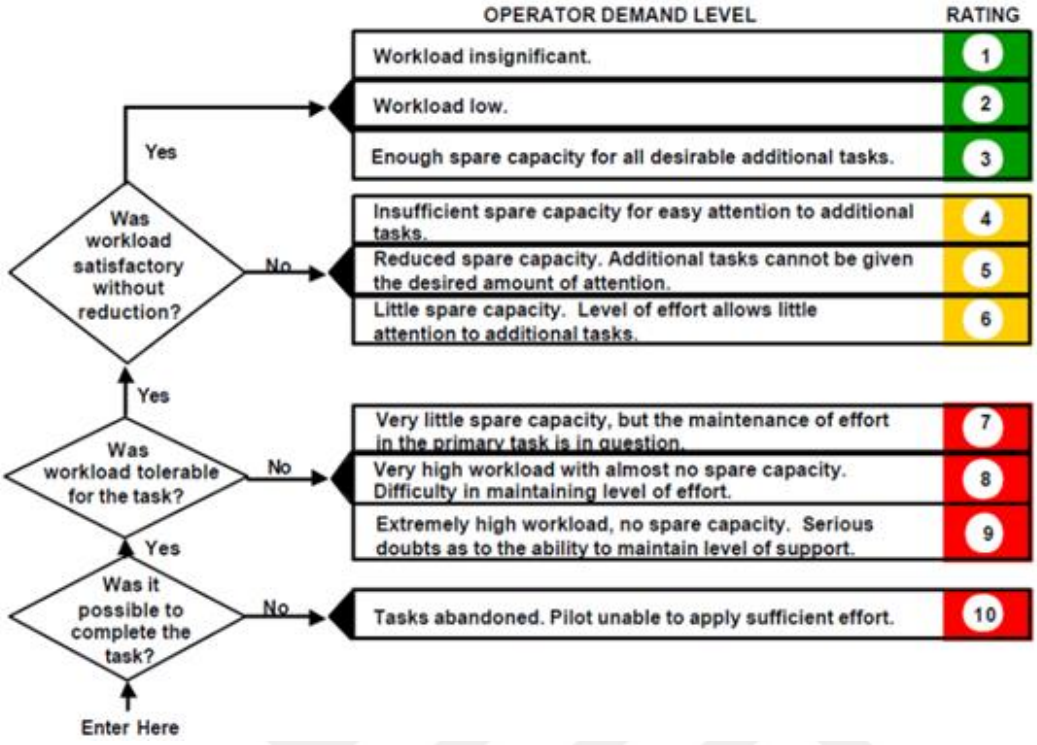

Figure 4: Bedford Workload Scale (Roscoe, 1984)

NASA Task Load Index

The NASA-TLX (Figure 5) is a multi-dimensional rating scale which is developed through laboratory studies (Hart & Staveland, 1988). It consists of six indicators to assess subjective workload which are mental demand (MD), physical demand (PD), temporal demand (TD), performance (OP), effort (EF) and frustration level (FR). Definitions of these indicators are given in Table 1.

NASA-TLX calculates an overall score (range 1-100) using six individual scale ratings (range 1-20) and their corresponding weights. After a user scores each of six indicators, the indicators are compared in pairs to determine their weights. The number of times an indicator is preferred in those pairwise comparisons determines the weighting of that indicator scale for a given task for the user. Then, the weighted sum of the indicators is divided by the number of paired comparisons to obtain a workload score between 0-100.

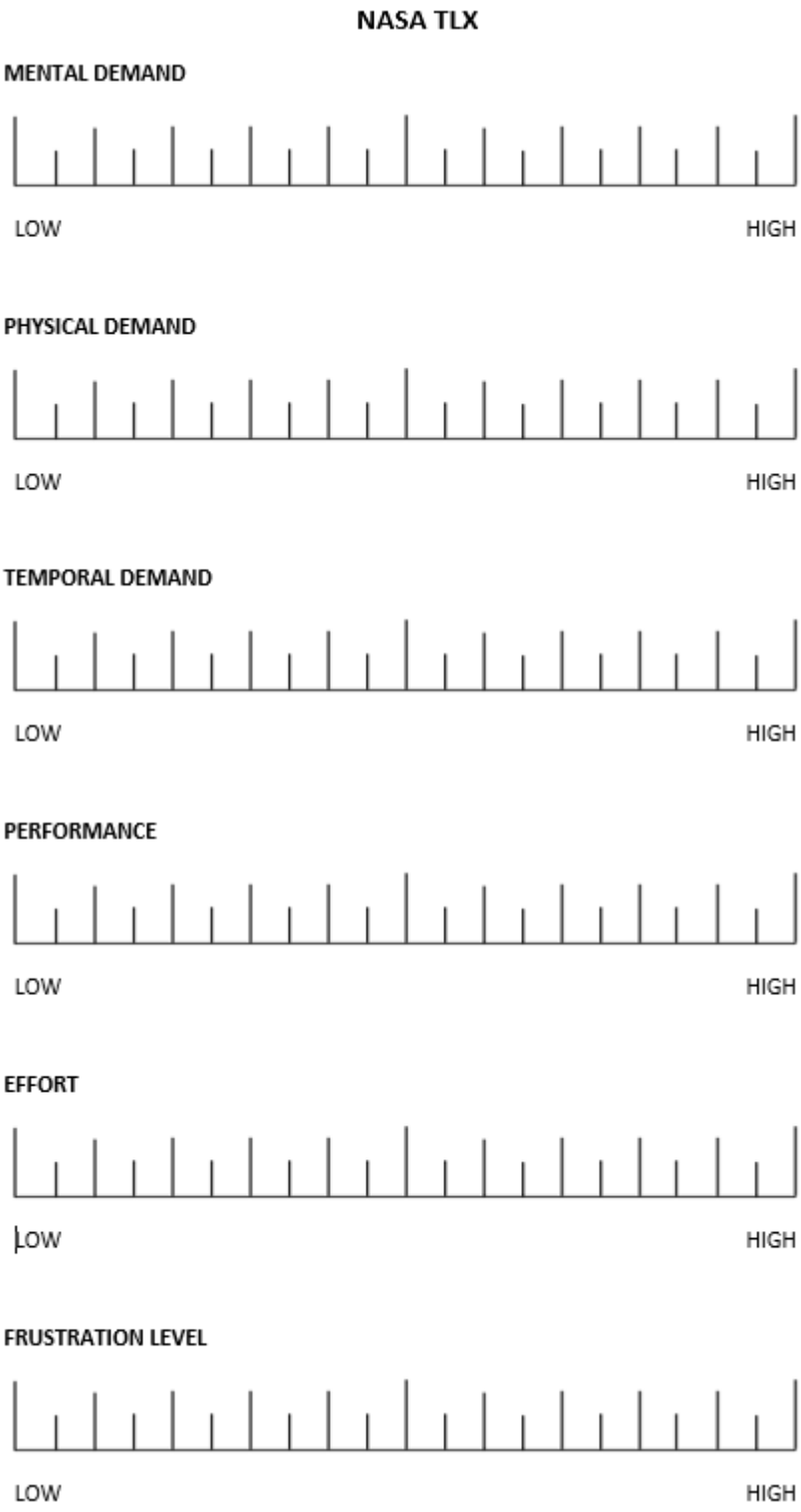

Figure 5: NASA-TLX Rating Scale (Hart & Staveland, 1988)

Table 1

Definitions of NASA-TLX indicators (Rubio, Díaz, Martín & Puente, 2004)

TITLE	ENDPOINTS	DESCRIPTIONS
MENTAL DEMAND	Low/High	How much mental and perceptual activity was required (e.g. thinking, deciding, calculating, remembering, looking, searching, etc.)? Was the task easy or demanding, simple or complex, exacting or forgiving?
PHYSICAL DEMAND	Low/High	How much physical activity was required (e.g. pushing, pulling, turning, controlling, activating, etc.)? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?
TEMPORAL DEMAND	Low/High	How much time pressure did you feel due to the rate or pace at which the task or task elements occurred? Was the pace slow and leisurely or rapid and frantic?
PERFORMANCE	Good/Poor	How successful do you think you were in accomplishing the goals of the task set by the experimenter? How satisfied were you with your performance in accomplishing these goals?
EFFORT	Low/High	How hard did you have to work (mentally and physically) to accomplish your level of performance?
FRUSTRATION LEVEL	Low/High	How insecure, discouraged, irritated, stressed, and annoyed versus secure, gratified, content, relaxed, and complacent did you feel during the task?

2.2.4. Physiological Measurements

Physiological methods are indirect measurements which are relational with cognitive load such as Electrocardiogram (ECG) that shows the heart rate activity and Electroencephalogram (EEG) that monitors brain activity (Brookhuis & De Waard, 2010), or eye tracking and pupillometry (Klingner, 2010). Behavioral methods analyze user's behavioral activities such as mouse tracking and body positioning (Elkin-Frankston, Bracken, Irvin & Jenkins, 2017). These methods attempt to analyze workload level based on physiological or behavioral changes. They do not require additional attempt from user to rate workload, data is monitored simultaneously while user performs experiments. Casner and Gore indicates that there is no powerful and rich theory behind physiological measurements (2010). They are accepted reliable because of their objectivity, but none of them alone can precisely capture the notion of workload. They require special measurement equipment to collect data.

2.2.5. Mixed Method Studies

Since cognitive models, physiological and psychometric measurement instruments have benefits and disadvantages that apply to different situations, previous research

has also combined multiple methods for more consistent and reliable measurement and estimations in mixed methods studies.

For instance, Zhang and colleagues (2015) integrate NASA-TLX, SWAT (Subjective Workload Assessment Technique) and VACP (visual, auditory, cognitive, psychomotor) to evaluate pilot workload. Their experiments are based on real flight tasks and conducted with 22 Airbus A320 crewmembers. They use NASA-TLX for total measurement analysis of workload. For the analysis of tasks, they use VACP model to pre-test and SWAT model to post-test. In addition, they use BCMS (Behavior-Cognitive Model Scale) to measure specific cognitive resources of pilots in each task. Rozado and Dunser (2015) combine EEG and pupillometry data to develop brain computer interface which monitors real time workload by using common average reference for data analysis. Klingner (2010) uses eye tracking and pupillometry data together to detect short-term changes in cognitive load based on scan paths analysis while performing visual tasks. Ramkumar and colleagues (2019) analyze GOMS and NASA TLX data in combination. They analyze the relations between these methods and propose HCI design suggestions based on their synthesis of the analysis results. Zheng and Jie (2019) use NASA TLX and eye blink rates for workload assessment. They compare NASA TLX results and eye blink rates both for flight simulator test and flight test. They imply that NASA-TLX results were significantly influenced by flight tests and environments, but eye blink rate only showed significant difference for environments. Because of the weak relation between these methods, they suggest more significant psychophysiological measurements. These studies, however, has not developed a model for predicting the workload or effort based on the measurements they combined.

2.2.6. Bayesian Models for HCI

Cognitive models such as GOMS provides a prior information about an average user. Then as we collect more data about the users, we can use this additional information to refine our prior information to have more accurate, user-specific information. In that regard, Bayesian methods offers a suitable approach in HCI studies to combine multiple sources of information, but Bayesian studies in HCI are still limited. Existing studies mostly focus on adaptive interfaces and emotion understanding.

Nguyen and Do (2009) indicated that the basis of an adaptive system is user model that includes personal information. They integrated a Bayesian model and an overlay model to infer user's knowledge by collecting data from the user during learning process (Nguyen and Do, 2015). Similarly, Rim and colleagues (2013) used Bayesian inference to predict the user's preferences according to the context on a Web interface.

Song and Cho (2013) created a context-adaptive user interface to manage a ubiquitous home environment which uses Bayesian network to predict the necessary devices and a behavior network to select the needed functions according to the situation. They showed that Bayesian network predicted user requirements efficiently and adaptive

user interface was more useful than fixed user interface. Conati and VanLehn (2001) designed an adaptive user interface based on Bayesian network to support the understanding of instructional material.

Huang and colleagues (2011) used Bayesian classification to design environmental monitoring interface which let users select and allocate factors on the interface freely and present them useful data about environmental quality variations. Ruokangas and Mengshoel (2003) constructed a unified Bayesian model to produce intelligent user interface by filtering complex weather information for pilots. Lu and colleagues (2015) created a Bayesian network which involves the head gesture statistic inference model and multi-view model (MVM) for head gesture recognition. Dudley, Jacques and Kristensson (2019) used Bayesian optimization for objective refinement of interface designs, and they especially reflected that crowdsourcing paired with Bayesian optimization can quickly and effectively support interface design in many cases.

Human emotions are important for HCI, because emotion has close relationship with human cognition and motivation (Akgun, Akilli & Cagiltay, 2011). Bayesian models have also been developed to identify human emotions in HCI. Gao and Wang (2015) developed a Bayesian model for emotion recognition from electroencephalogram (EEG) signals which handles specificity and generality of emotions in parallel. Moreover, Ko and Sim (2009) developed facial expression recognition system by using six universal emotional categories based on Bayesian network.

In our literature review, we could not find Bayesian studies that estimate mental workload and performance in user interfaces. However, there were studies that estimated workload in the use of construction machines and helicopter. Luo and colleagues (2019) developed models to estimate human workload while performing teleoperation tasks by analyzing physiological data in the terms of Bayesian approach. They conducted experiments to get human gaze trajectory and pupil size data while teleoperating of an unmanned high mobility multipurpose wheeled vehicle in parallel with performing a secondary task. They combined this data to make real-time workload assessment based on Bayesian inference approach. Besson and colleagues (2013) also studied on a model to estimate helicopter pilots' workload based on Bayesian network. Besson and colleagues (2013) also studied on models to estimate helicopter pilots' workload based on Bayesian network. They conducted experiments both in laboratory environment which has low ecological validity and in a full-flight simulator to collect physiological data. They collected subjective data with NASA-TLX rating scale at the end of each task, too. Then, they developed models for laboratory and virtual reality environment to estimate pilot's workload based on Bayesian network.

In summary, previous research on the use of Bayesian methods in HCI primarily focused on adaptive user interfaces and user emotions. Mental workload and performance prediction is a suitable domain for the use of Bayesian methods. Combining knowledge provided by cognitive models such as GOMS, and data collected from users can lead to accurate and personalized prediction of workload

without needing to collect large amounts of data. The potential benefits of Bayesian workload estimation include better understanding workload on a personalized basis, and decreasing the cost of data collection for this task. Despite these potential benefits, previous research on workload estimation in HCI has not focused on Bayesian methods.

CHAPTER 3

METHODOLOGY

This study aims to develop models that estimate cognitive load and performance based on a Bayesian data analysis approach. These models revise the predictions obtained from a cognitive model with the data of subjective scales and task execution times. This chapter will describe the proposed Bayesian models, and the methodology followed in the case studies.

3.1. Bayesian Models

We developed two types of Bayesian models with different levels of complexity. The first type updates the workload and task completion time estimates obtained from Cogulator based on Bedford scale and task execution time data without accounting for the differences between the users. The second type also considers the differences between the users and takes that into account while making predictions. Each of these model types have been instantiated for estimating mental workload and estimating task completion times. All Bayesian models have been implemented in WebPPL (Goodman and Stuhlmüller, 2014) which is a probabilistic programming language based on JavaScript.

3.1.1. Type 1 Workload Model

This model has three types of parameters: *observedWorkload*, *taskWorkload* and *taskSigma*. We defined *observedWorkload* with a Gaussian distribution that gets *taskWorkload* as its mean and *taskSigma* as its standard deviation. This parameter indicates the observed workload in an experiment.

$$observedWorkload \sim \text{Gaussian}(taskWorkload, taskSigma)$$

We defined *taskWorkload* as Gaussian distribution which gets Cogulator workload value as mean with fixed standard deviation 1. This parameter represents prior distribution of workload estimation.

$$taskWorkload \sim \text{Gaussian}\ (cogulatorEstimate, 1)$$

We defined *taskSigma* as Gamma distribution as given below which represents the standard deviation of workload distribution.

$$taskSigma \sim \text{Gamma}(1, 1)$$

Figure 6 shows the DAG representation of simple workload estimation model. After we defined all variables, we instantiate *observedWorkload* from each experiment with the Bedford workload data collected and update the distributions *of taskWorkload* and *taskSigma* by the MCMC method.

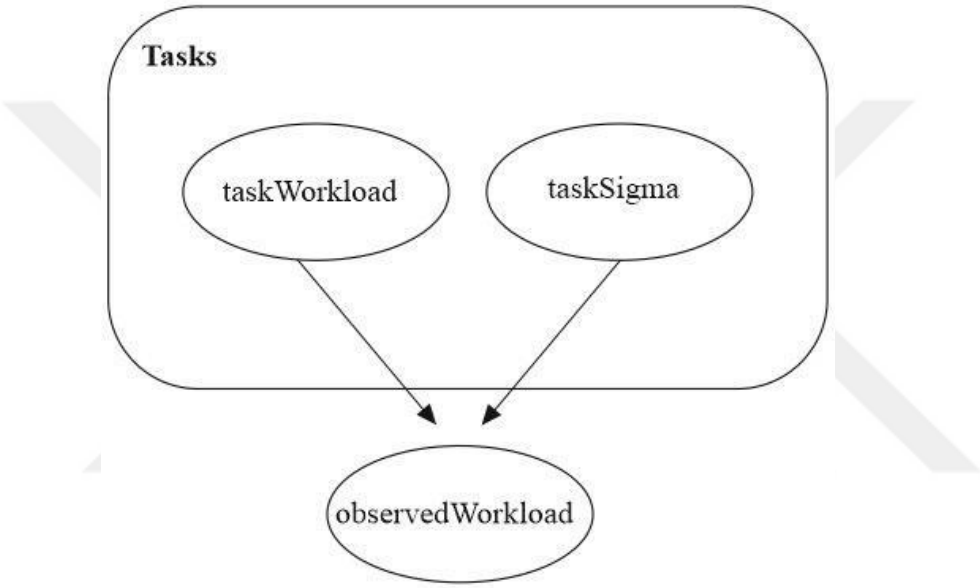

Figure 6: Type 1 Workload Model

3.1.2. Type 1 Execution Time Model

We also adapted the Type 1 to analyze real execution time based on the predictions provided by Cogulator and update estimation of execution time. Parameters in this model are observedExcTime, taskExcTime and taskSigma. We defined observedExcTime as Gaussian distribution that gets taskExcTime as mean and taskSigma as standard deviation. This parameter represents observed times in experiments.

$$observedExcTime \sim \text{Gaussian}(taskExcTime, taskSigma)$$

We defined taskExcTime as Gaussian distribution which gets Cogulator time estimation value as mean with a fixed standard deviation of 15. This parameter represents prior distribution of execution time estimation.

$taskExcTime \sim \text{Gaussian}\ (timeCogulator, 15)$

We defined taskSigma as Gamma distribution which represents the standard deviation of posterior execution time distribution.

$taskSigma \sim \text{Gamma}(2, 1)$

Figure 7 is the DAG representation of simple time estimation model. After building the model we instantiate *observedExcTime* with the execution time data recorded in the experiments and model updated *taskExcTime* and *taskSigma* distributions by the MCMC method.

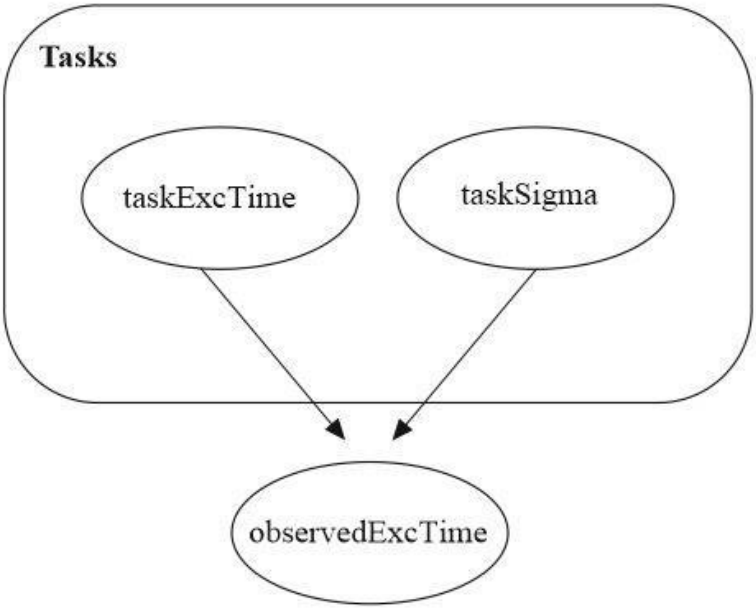

Figure 7: Type 1 Execution Time Model

3.1.3. Type 2 Workload Model

Type 2 model also accounts for the difference between users and contains six type of parameters (see Figure 8). In this model, *observedWorkload* represents observed workload time for a particular task and user based on the collected Bedford scale data. It is modelled with a Gaussian distribution. The mean of this distribution is defined by the average workload of a task plus the relative skill of the user. The relative skill of the user is defined by how many standard deviations that the workload of a particular user is different from the average workload. The standard deviation of the *observedWorkload* is defined by *taskSigma* which is modelled with a Gamma distribution.

$$observedWorkload \sim \text{Gaussian}\ (taskWorkload + userTaskSkill * taskSigma, taskSigma)$$

$$taskSigma \sim \text{Gamma}(1, 1)$$

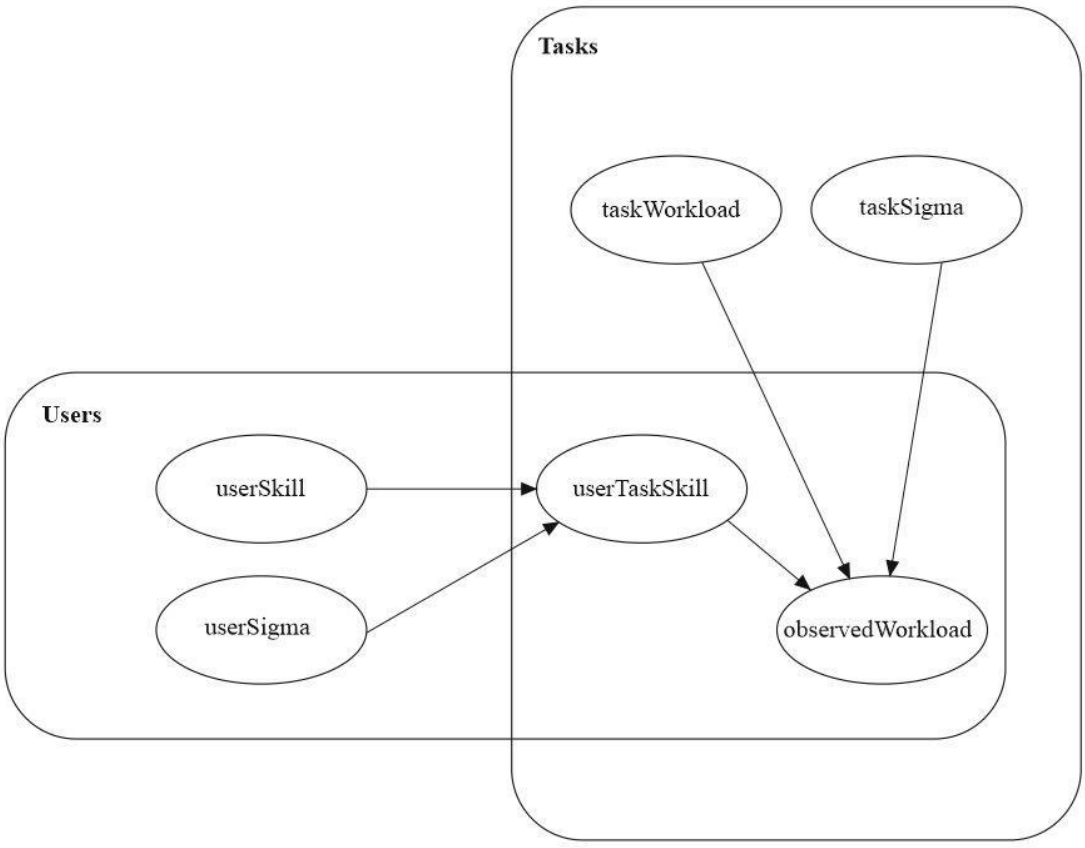

Figure 8: Type 2 Workload Model

The average workload of a task is represented by *taskWorkload* which is modelled with a Gaussian distribution. It gets Cogulator workload estimation value as mean with fixed standard deviation 2 and represents prior distribution of workload estimation.

$$taskWorkload \sim \text{Gaussian}\ (workloadCogulator, 2)$$

The *userTaskSkill* parameter represents the relative workload of a user with respect to other users for a particular task. In other words, this variable represents how many standard deviations this user is away from the mean for a particular task. The average skill of a user for all tasks is represented by *userSkill*. We assign a prior mean of 0 and a standard deviation of 1 for this parameter. The last parameter is *userSigma* which

represents the variation between the tasks for a user. We assign a Gamma prior for this parameter.

$userTaskSkill \sim \text{Gaussian}\ (userSkill, userSigma)$

$userSkill \sim \text{Gaussian}\ (0, 1)$

$userSigma \sim \text{Gamma}(1, 1)$

After building this model, we instantiate *observedWorkload* for each experiment with the Bedford workload data and revise the distributions of other variables by the MCMC method.

3.1.4. Type 2 Execution Time Model

We also adapted Type 2 model to estimate task execution times accounting for the differences between the users (Figure 9).

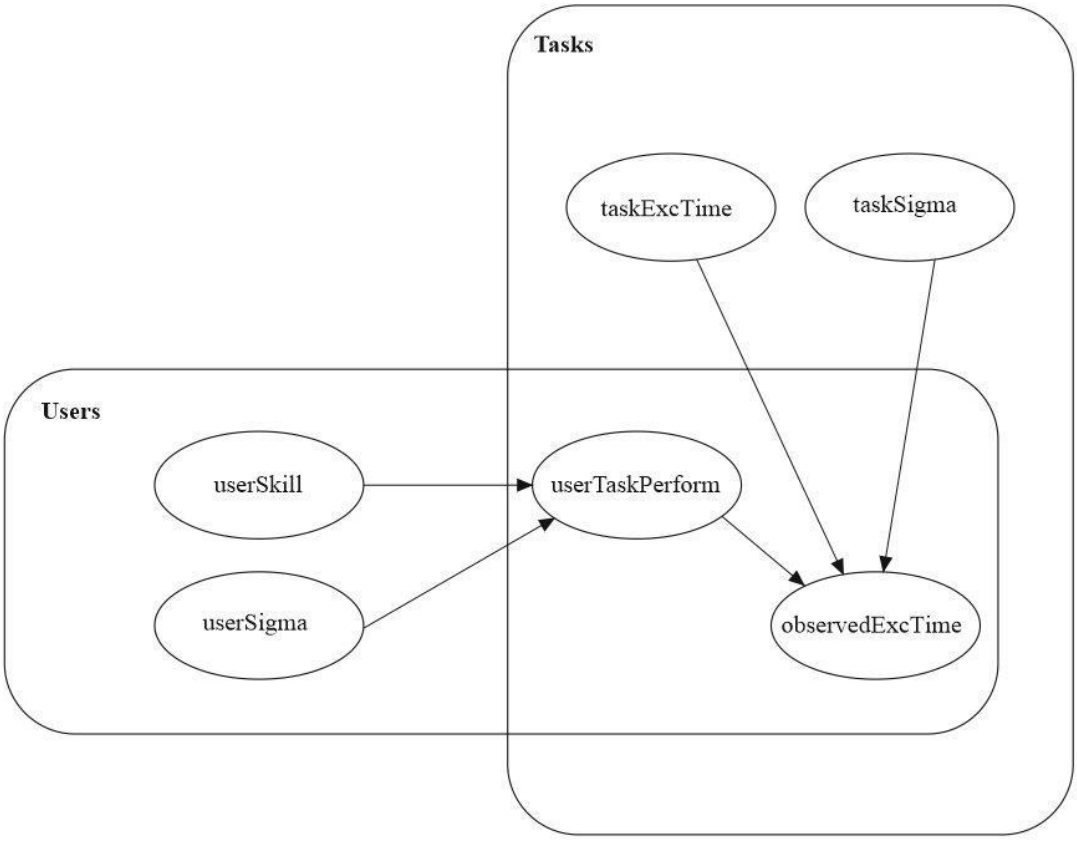

Figure 9: Type 2 Execution Time Model

In this model, *observedExcTime* represents observed execution time for a particular task and user. It is modelled with a Gaussian distribution. The mean of this distribution is defined by the average execution time of a task plus the relative performance of the user, i.e. how many standard deviation away that particular user is from average task execution time. The standard deviation of the *observedExcTime* is defined by *taskSigma* which is modelled with a Gamma distribution.

$$observedExcTime \sim \text{Gaussian}\ (taskExcTime + userTaskPerform * taskSigma, taskSigma)$$

$$taskSigma \sim \text{Gamma}(2, 1)$$

The average execution time of a task is represented by *taskExcTime* which is modelled with a Gaussian distribution. It gets Cogulator execution time estimation value as mean with fixed standard deviation 10 and represents prior distribution of workload estimation.

$$taskExcTime \sim \text{Gaussian}\ (timeCogulator, 10)$$

The *userTaskPerform* parameter represents the relative performance of a user with respect to other users for a particular task. In other words, this variable represents how many standard deviations this user is away from the mean execution time of a particular task. The average skill of a user for all tasks is represented by *userSkill*. We assign a prior mean of 0 and a standard deviation of 1 for this parameter. The last parameter is *userSigma* which represents the variation between the tasks for a user. We assign a Gamma prior for this parameter.

$$userTaskPerform \sim \text{Gaussian}\ (userSkill, userSigma)$$

$$userSkill \sim \text{Gaussian}\ (0, 1)$$

$$userSigma \sim \text{Gamma}(1, 1)$$

After building this model, we instantiate *observedExcTime* for each experiment with the execution time data collected from each experiment and revise the distributions of other variables by the MCMC method.

3.1.5. NASA-TLX measurements and Bayesian Models

We considered incorporating NASA-TLX measures alongside Bedford scale measures to the Type 2 Workload model described in Section 3.1.3. For a preliminary analysis, we performed a linear regression analysis between the Bedford scale measures and NASA-TLX measures. We observed that, the strength of relation between these two types of measurements were low hence we did not include NASA-TLX to our model. Results of this regression analysis is shown in Section 4.1.4.

3.2. Case Study

We applied the method described above and the resulting BN model to two user interfaces in the defense industry domain. The first one was a tank driver interface and second one was torpedo counter measure system. Five different scenarios were designed for each interface and the user was asked to perform certain tasks for these scenarios. Data collection was performed using subjective workload scales. Adapted Bedford workload scale was applied at the end of each task, and the Nasa TLX rating scale was applied for general evaluation after all tasks of an interface were completed, in addition, execution time of each task was recorded.

3.2.1. User Interfaces

Tank Driver System

The tank driver system has a simple interface. It has clickable controls where selections are made. In addition, there are sub-menu fields for data entry. Apart from these controls and menu items, it is not much different from a normal navigation screen. Since it is only the interface that the driver uses, it does not have complex capabilities such as fire control. So, almost every interaction is defined in this system.

The user performs operations such as IR/TV camera switching, front/rear camera switching, night/day mode switching. Moreover, the operator may need to enter text-based data such as destination information or location information when necessary. When there is any system error or warning, the details appear and disappear on the screen for a certain period of time. The driver may need to act according to these stimuli and change some settings. It can also receive voice commands from the commander. In the light of this information, the tank interface tasks can be summarized as follows.

The first task contains "look - point -click" subtasks such as "change mode", "switch to rear camera", and entering some data from the experiment instruction to the relevant fields in the interface.

The second task consist of similar subtasks, but needed information for data entry is partial in instruction page. Missing information comes from the commander verbally, and driver has to memorize it for a while. After completing previous tasks, driver has to recall that information to enter.

Third task is similar to second one, this time driver gets two missing information from commander and has to memorize and recall more chunks.

Similarly, in the fourth task, driver gets all three different information from commander and completes the task.

In the fifth task, after entering the whole data to the system, driver gets an error for one value which says “This value is not suitable for the system, try double it” and disappears. Other entered information also disappears on the screen. While performing multiplication in the head and entering the relevant data, driver needs to remember and enter the other two data connected once again.

Torpedo Counter Measure System

User interface of the torpedo counter measure system is more complex. It has more menu items that are clickable and more sub-menu fields for data entry. In addition, most of the screen is reserved for the part that we call the tactical display. Tactical display reflects the system location and orientation, and traces (see Figure 10). The small circle around the system represents the critical area, while the large circle represents the entire defended area. There can be three different type of traces. Red trace represents danger, yellow trace represents potential danger which can be danger but not classified yet, and blue trace represents insignificant traces that are not in danger or warning class. When we click on a trace, we get detailed information about it such as depth or bearing data in another part of the interface reserved for this.

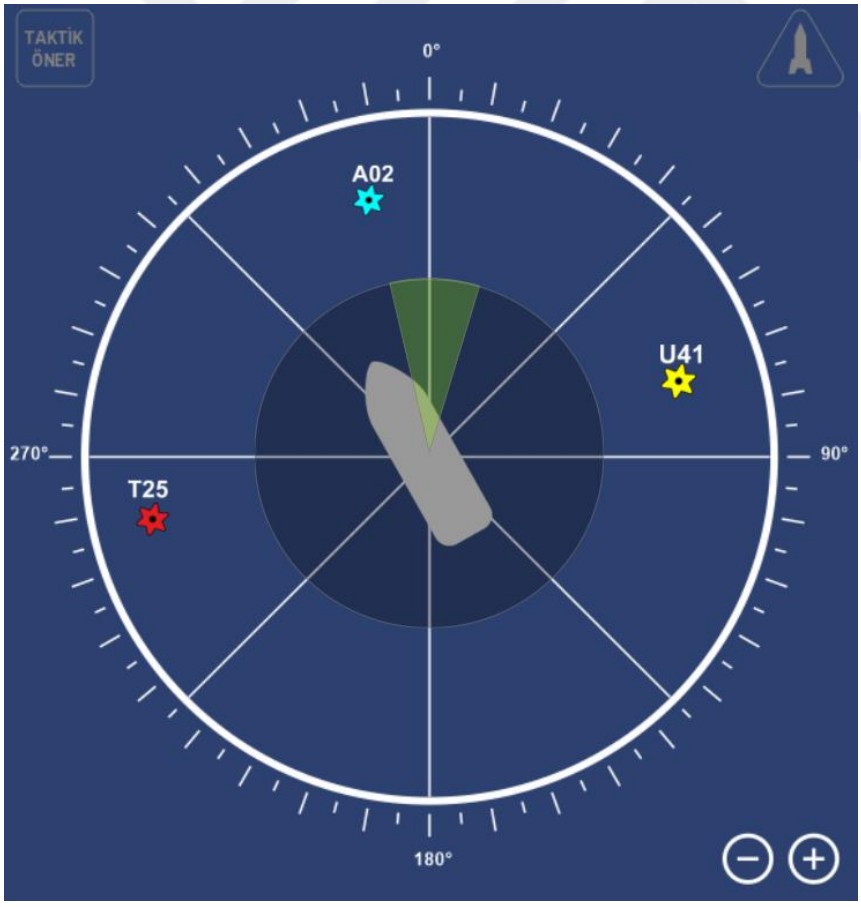

Figure 10: Tactical Display

Tasks related to this interface are designed assuming that the system is in fully manual mode and tactics are carried out one by one. Similarly, here too, voice orders may come from the commander. We used a simulation program developed by ASELSAN to create random traces in scenarios by identifying classification and number information like 4 insignificant, 2 classified as danger, and 3 classified as potential danger. Every single trace on this screen has identity such as "A02" and we can also set their speed in simulation program. In the light of this information, tasks can be summarized as follows.

The first task contains "look - point -click" subtasks like switching to operational mode and clicking on a requested trace to see detail information about it. This task requires to memorize one random data of requested trace to recall later. Then it requires to perform simple arithmetic operations by giving warnings such as; “This value is not enough, try multiplying it by 4!” on data entry stage. There are also verbal orders from commander in this scenario.

The second task of this system exactly contains identical steps with the first one. Only difference is the screen complexity. In the first scenario there are many extra traces on the screen, while in the second there are only 2 different colored traces, apart from the trace that the user was told to follow. Thus, when the command comes from the commander to follow the relevant trace, it is easier to find the relevant trace in the 2nd scenario.

Third task contains similar subtasks with previous ones, but this time operator simultaneously checks if the critical area is safe while following the requested trace and performing other requested subtasks. If there is a trace entering the critical area operator must verbally say the trace identity to commander while keep performing other duties.

Similarly, fourth scenario contains identical subtasks with third one and only difference is screen complexity. While there are numerous traces on tactical display in the third scenario, there are only 3 traces in this one.

In the last scenario, the operator observes the critical area while performing certain tasks. What to do in this scenario is always communicated verbally and definitively by the commander. Here, the user performs many sequential and simultaneous subtasks on many all traces on the screen under time pressure according to the commander’s verbal instructions.

3.2.2. Cognitive Models

We used Cogulator to derive our prior estimates for execution time and workload. Cogulator provides a time estimate in seconds and a workload estimates in 1 - 10 rating scale for a modelled task (see Section 2.2.2 for a detailed description of Cogulator).

Each task were modelled via Cogulator to derive prior data of execution time and workload.

Cogulator models were built by the author who is an experienced UX/UI engineer and have been involved in the design of the interfaces described in Section 3.2.1. Two interfaces that differ in their relative complexity were selected, five task scenarios with increasing level of complexity were designed for each interface as described in Section 3.2.1. Each scenario were performed by the author and its operations were examined iteratively to accurately identify the associated sequence of operations. Think operations have been added when an information is recalled from memory or when an order requiring an arithmetic operation is made. Cogulator also assisted identification of operations by giving warnings such as "Hands are not on keyboard" when the type operator is used, or "Hands are not on the mouse" when the click operator is used. Default time estimates from Cogulator were used for all operations except look, think and recall operators in the models. The Cogulator models for first tasks of interfaces are given in the Appendix.

Note that, Cogulator's predictions can be interpreted as estimations for an average expert user. It cannot account for hardware differences such as the size of the screen, whether there is more than one screen, the use of a trackball instead of a mouse, or personal differences such as cognitive capacity differences, multitasking ability, or any disability may affect the basic assumptions. Therefore, Cogulator's predictions are used as prior time and workload estimations in our Bayesian model, which are revised based on data about subjective workload assessments and the actual user execution times.

3.2.3. Data Collection

In the beginning of the experiments, participants were given a form of consent, and the experimenter described the purpose briefly. Afterwards, participants were shown a page that describes the steps of the experiment and introduced the subjective workload instruments that are used in the experiment. Before, starting the experiments with each interface, a sample task was shown to the participants to introduce them the interface. After completing the sample task, each task scenario was described to the participants separately, and scenarios were run when the participants indicated that they are ready. The tasks were presented to each user in the same order. The experiment was concluded by thanking the participant and asking for feedback to the experimenter. Each experiment took approximately 30 minutes. They were run on the same computer at ASELSAN.

During the experiments, we collected the execution time from the users, and we asked users to complete the adapted Bedford workload scale after completing each task and NASA TLX rating scale after completing all tasks for each interface. Since Bedford Workload Scale is unidimensional and it is suited for complex tasks with high cognitive requirements. We collected NASA-TLX data for overall evaluation of our

user interfaces as it provides a more thorough multidimensional assessment of workload by asking the user to compare multiple dimensions to estimate their values and weights (see Section 2.2.3 for a detailed description of Bedford workload scale and NASA TLX).

3.2.4. Participants

Twenty people, with age, ranged 23-40, participated to the experiment. The average age of participants was 28.75 years. 13 of which were male and 7 of which were female. All participants were volunteers for the experiment and were able to leave whenever they want. All of them were ASELSAN employees who were familiar with defense user interfaces. Among the participants, 7 of them were expert users who are system engineers and more familiar with the systems. 13 of them were from design teams including mechanical, industrial, and software design.

3.2.5. Analysis Procedure

In order to analyze the performance of the proposed Bayesian models in the case studies, we have built the Bayesian models described in Section 3.1 and populated the *timeCogulator* and *workloadCogulator* priors in those models with the estimates obtained from the cognitive models described in Section 3.2.2. We have collected data about workload and execution times as described in Section 3.2.3, and entered this data to the *observedWorkload* and *observedExcTime* parameters in the Bayesian models.

Using Type 2 models, we analyzed the difference between prior (Cogulator) estimates and posterior estimations of workload and execution times to identify changes and cognitive resource requirements in the tasks. We also compared users' cognitive abilities and performance relative to the average in our model.

We also analyzed the predictive performance of Type 2 models and compared it with the predictions provided by the Cogulator. We divided the data into training and test sets with 80% to 20% ratio. We estimated the posteriors for workload and execution times of each user for each task using the training set. Afterwards, we compared the posterior execution times and workload for the test set with the true values. We also compared the predictions of Cogulator for the test sets. Mean Absolute Error was used as a summary metric for predictive accuracy.

Finally, we also made a linear regression analysis between NASA-TLX and Bedford measurements as described in Section 3.1.5. The following section presents the results of these analyses.

CHAPTER 4

ANALYSIS AND RESULTS

This chapter presents the results for the case study. We examined the posteriors of tasks and the difference between the users in terms of workload and time. In addition, we assessed the predictive performance of workload and time estimation models by dividing the data into training and test sets and assessing the predictive performance in the test set.

In Section 3.1, we proposed two types of Bayesian models; Type 1 models estimate the workload and execution time of tasks, and Type 2 models expand them by accounting for the difference between users. We used Type 2 models for all of the analyses presented in this section.

The tasks associated with each interface are numbered from 1 to 5. Detailed information about interfaces and tasks are given in Section 3.2.1.

4.1. Workload Estimation

We used Type 2 Workload Model (see Figure 8) described in Section 3.1.3 to analyze tasks, users and model's predictive performance based on workload. In addition, we applied regression analysis for Bedford and NASA-TLX measurements.

4.1.1. Analysis of Tasks

In this section, we examine the posterior workloads of 10 tasks performed in two different interfaces. Here, our prior workload estimation values come from Cogulator model we developed. After we enter observed workload data collected from users through experiments (see Section 3.2.3 f), our model updates workload data with MCMC method and we get posterior workload estimation distributions. Prior workload estimation from Cogulator and posterior workload estimation of our model for each task of Tank interface are presented in Figure 11.

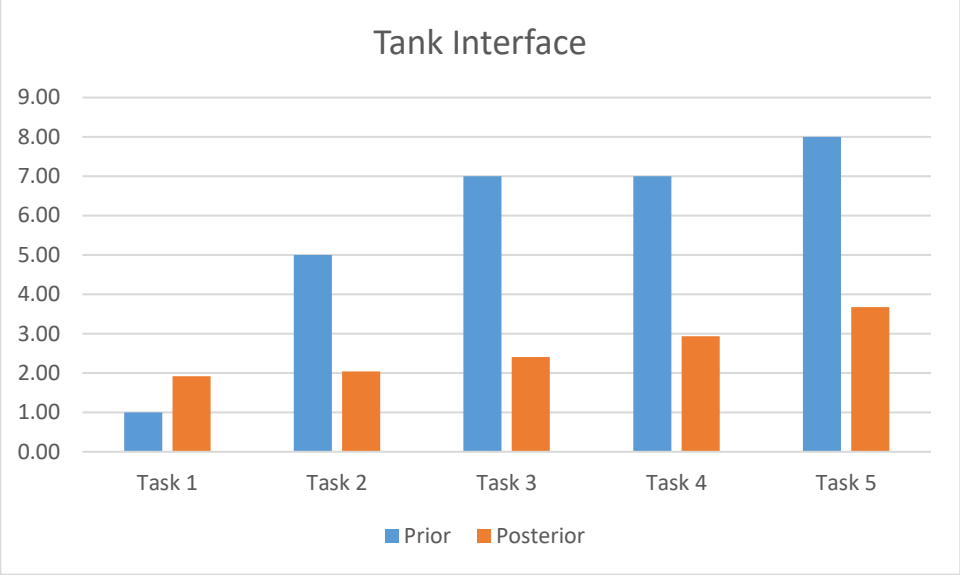

Figure 11: Prior and Posterior Workload of Tank Interface Tasks

According to the Cogulator predictions (priors), for our tank interface, it was the 1st task that should be performed the easiest, it was also 1st task according to our model. Similarly, the task with the highest workload is the 5th task in both. The 1st task consists of only "look, point, and click" steps, and it consists of entering some values written in the instruction page into the relevant fields on user interface. Other tasks also require memorizing some values from user and information chunks increase in every task from 2 to 5 as expected. Note that the values of prior and posterior workloads are quite different. The Cogulator estimates for the first task and other tasks differed considerably, whereas posterior workloads of those tasks were closer. While 1st task has the lowest update rate with 92 percent change, 3rd task has the highest change with a rate of 459 percent. The posterior probability distributions of 1st task that has lowest workload and 5th task that has highest workload in this interface are presented in Figure 12 and 13.

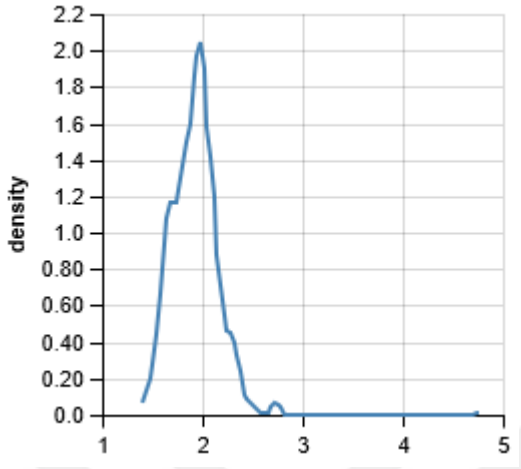

Figure 12: Task-1 Workload of Tank

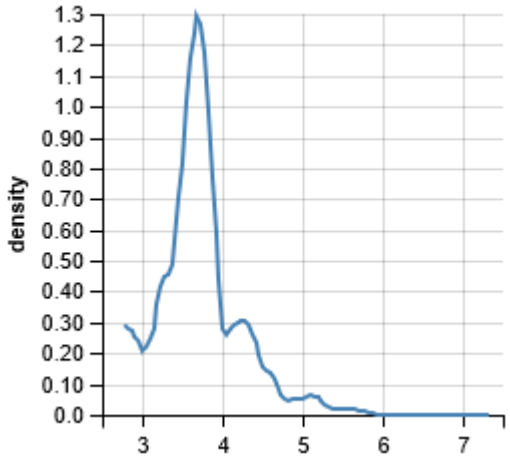

Figure 13: Task-5 Workload of Tank

Table 2

Task Workload Order in Tank Interface

Rank	Cogulator	Model
1	Task 5	Task 5
2	Task 3, Task 4	Task 4
3	Task 2	Task 3
4	Task 1	Task 2
5		Task 1

The workload of tasks in the first interface are ranked in decreasing order in Table 2. While the tasks with highest and lowest workload values are same according to Cogulator and our model, the orders of Task 3 and Task 4 is different. While Cogulator estimates the same workload value for Task 3 and Task 4, our model estimates a higher value for Task 4. It is expected, because in Task 4 scenario, user needs to memorize one more information chunk in working memory that comes from commander than Task 3.

For the TCMS (Torpedo Counter Measure System) interface, the prior workload estimation from Cogulator and the posterior workload estimation of our model for each task are presented in Figure 14.

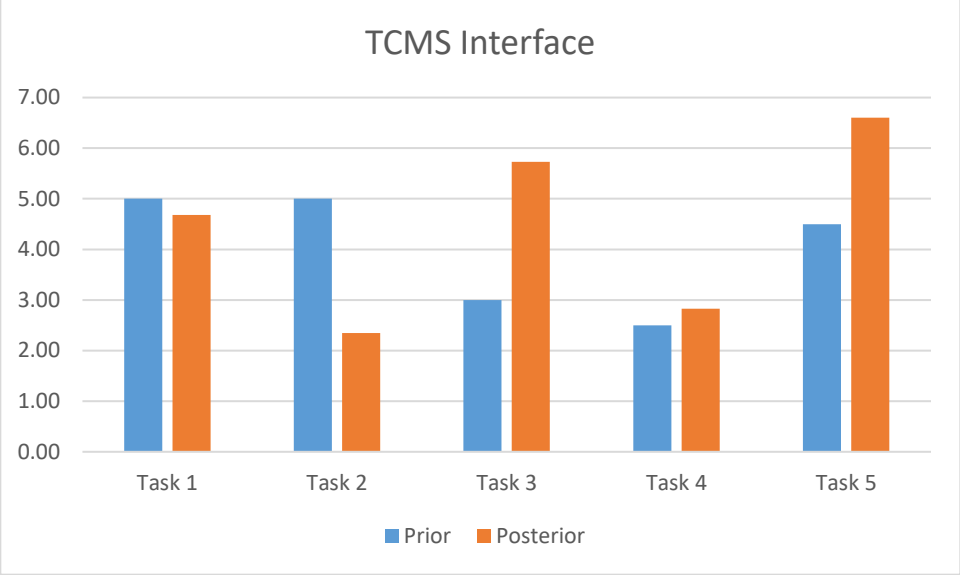

Figure 14: Prior and Posterior Workload of TCMS Interface Tasks

The differences between the order of the prior and posterior workloads are higher in this interface. While Cogulator predicts that the 4th task has the lowest workload and 1st and 2nd task has the highest workload. The posteriors revised by our model shows that the 2nd task has the lowest cognitive load and 5th task has the highest cognitive load (see Figures 15, 16, 17, 18 for the posterior probability distributions for these tasks). The 2nd and 4th tasks have similar sub-tasks like following a particular trace on screen. Number of traces on screen are also equal in these scenarios. But, while user only performs duties according to the commands in the 2nd one, 4th task also requires information from user verbally. The user has to give identity information of a trace verbally to the commander if it enters to the critical area. There is also randomness here, because user doesn't know if there will be a trace in critical area, so it is necessary to check critical area continually. But we don't have many traces on screen in these scenarios, so it is not difficult to check if there is a danger in critical area for 4th task. Performing sequential duties according to the commands in 4th scenario can be more challenging for user. In this interface, the largest change between the priors and posteriors was for the 3rd with 273 percent. The 1st task has the lowest change with 32 percent.

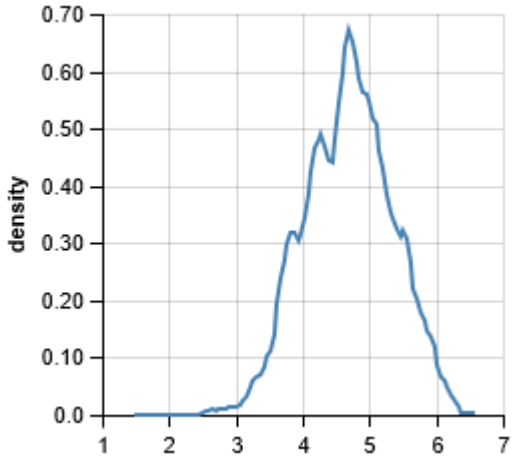

Figure15: Task-1 Workload of TCMS

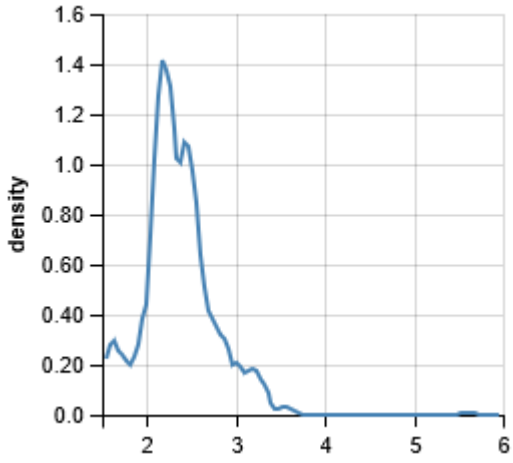

Figure 16: Task-2 Workload of TCMS

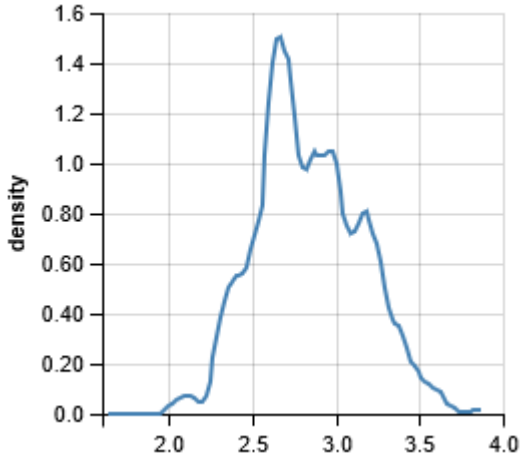

Figure 17: Task-4 Workload of TCMS

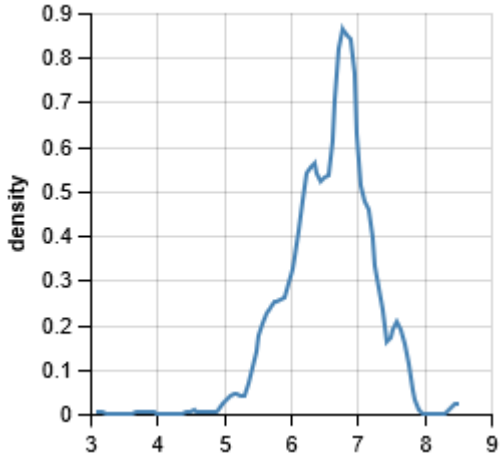

Figure 18: Task-5 Workload of TCMS

Prior and posterior workload of all tasks in the second interface are ranked in decreasing order in Table 3. Cogulator predicts equal cognitive load for 1st and 2nd tasks as they have identical sequence of operations. However, while in the 1st task there are many colored traces on the screen, in the 2nd task there are only two different colored traces, apart from the trace that the commander told to follow. Thus, when a command comes from the commander to follow the relevant trace, it is easier to find that trace in the 2nd scenario and simultaneously follow it to see if the critical area is safe. Similarly, there are many confusing traces in the 5th scenario, moreover this task contains more interaction steps under time pressure requiring more effort than others.

Table 3

Task Workload Order in TCMS Interface

Rank	**Cogulator**	**Model**
1	Task 1, Task 2	Task 5
2	Task 5	Task 3
3	Task 3	Task 1
4	Task 4	Task 4
5		Task 2

4.1.2. Analysis of Users' Task Skills

We examined the relative differences between the users by using the *userSkill* variable in the Type 2 model (Section 3.1.3). After we enter observed workload data for each experiment, our model updates *userSkill* variable and it represents how many standard deviations the user is away from the mean for a particular task. Figure 19 shows the distribution of this data.

According to this data; the task skills of 12 users out of 20 is below average because they need more task skill than mean, and 8 of them are above average because they need less task skill than mean. While 13th user (Figure 20) is at the top of the graphic with most task skill necessity, 10th user (Figure 21) is at the bottom with less skill necessity. This means that 13th user is the least skilled one while 10th user is the most skilled one. The most skilled three users according to this data are user 10, user 16 and user 19. In contrast, least skilled three users are user 6, user 8 and user 13.

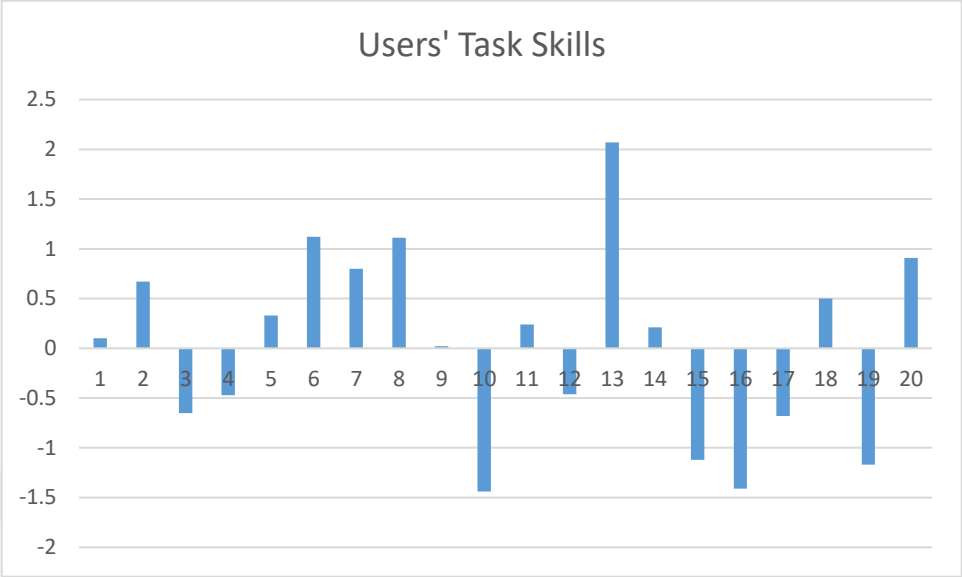

Figure 19: Task Skills of Users

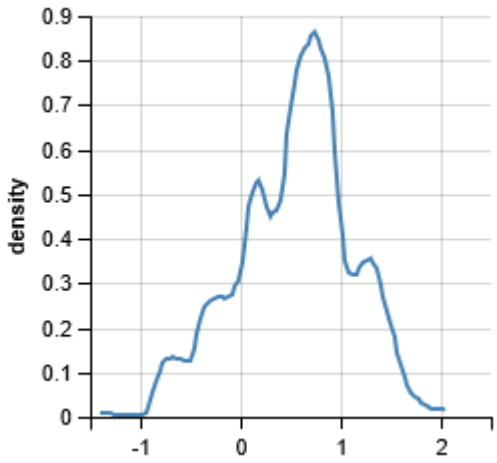

Figure 20: Task Skill of 13th User

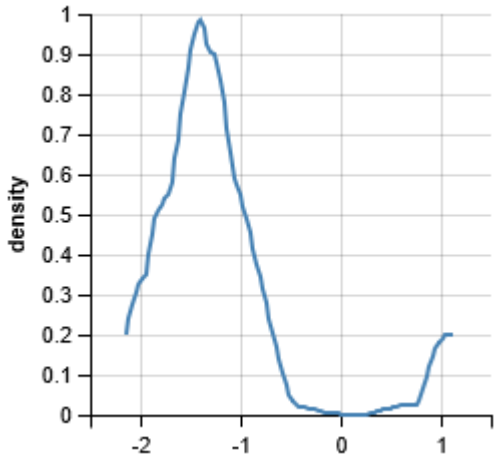

Figure 21: Task Skill of 10th User

4.1.3. Analysis of Model's Predictive Performance

For our Type 2 workload model, we randomly divided data into training and test sets with 80% to 20% ratio to analyze the model's predictive performance. We estimated the posterior workload of each user for each task based on the training set. We then compared the prior workload data of Cogulator and the posterior workload data of model with true values (Figure 22). While mean absolute error is 2.97 for Cogulator, it is 1.20 for our model according to the results.

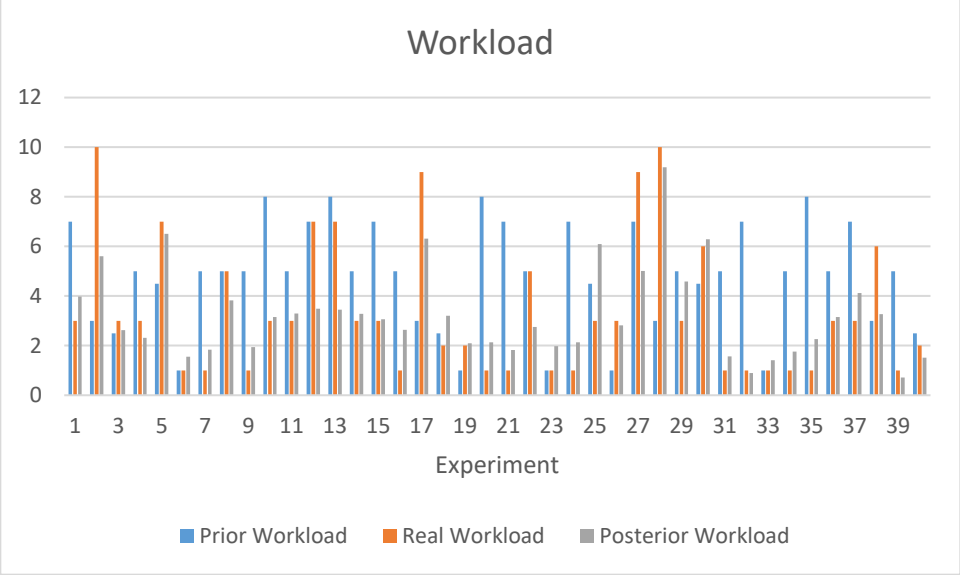

Figure 22: Prior, Real and Posterior Workload

4.1.4. Analysis between NASA-TLX and Bedford Scales

We used multiple linear regression analysis between Bedford scale measurements of each task of an interface and NASA-TLX value for the whole interface as a preliminary analysis to examine whether to include of NASA-TLX measurements in the Type 2 workload model. The aim of this analysis was to assess whether NASA-TLX measurement for the whole interface could be predicted with Bedford scale measurements for different tasks of that interface.

For the tank interface; a significant regression equation was not found ($F = 1.792$, $p < 0.179$), with an R^2 of 0.390. The regression equation coefficients were as follows but none of the coefficients except the intercept were statistically significant.

$$NASATLX = 22.667 + 2.120\,(Bedford_5) + 0.445\,(Bedford_4) + 1.703\,(Bedford\_3) - 2.244\,(Bedford_2) + 3.541\,(Bedford_1)$$

For the TCMS interface, a significant regression equation was also not found ($F = 1.954$, $p < 0.149$), with an R^2 of 0.411. The regression equation coefficients were as follows but none of the coefficients except the intercept were statistically significant.

$$NASATLX = 44.913 + 0.865\,(Bedford_5) + 1.740\,(Bedford_4) + 0.892\,(Bedford_3) - 1.861\,(Bedford_2) + 2.165\,(Bedford_1)$$

Based on these results, we did not include NASA-TLX values in the Type 2 workload model.

4.2. Time Estimation

We used Type 2 Execution Time Model (see Figure 9) proposed in Section 3.1.4 to analyze tasks, users and model's predictive performance based on execution time.

4.2.1. Analyses of Tasks

We compared the execution time predictions from Cogulator to the posterior execution times revised by our mode for 10 tasks performed in tank and TCMS interfaces. Prior workload estimation of Cogulator and posterior workload estimation of our model for each task of the Tank interface are presented in Figure 23.

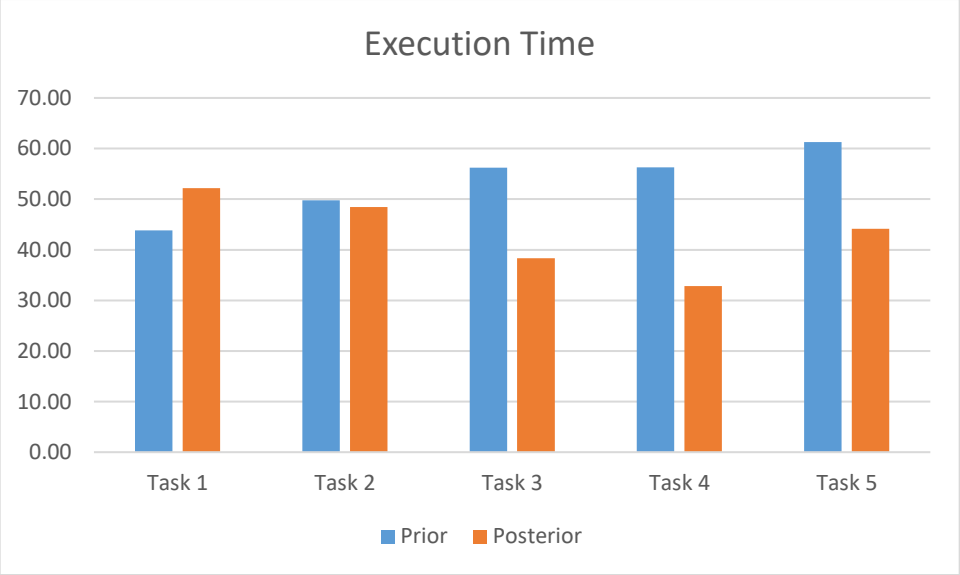

Figure 23: Prior and Posterior Execution Time of Tank Interface Tasks

For the tank interface, Cogulator estimates that the 1st task that should be performed the fastest, whereas the posteriors of the Bayesian indicated that 4th task was the fastest. In the 1st task, the data from the test paper is entered into the relevant places in the interface by the user, while in the 4th task, all of these data are verbally expressed by the commander and entered into the system by user under time pressure. The posterior probability distributions of 1st and 4th tasks are presented in Figure 24 and 25.

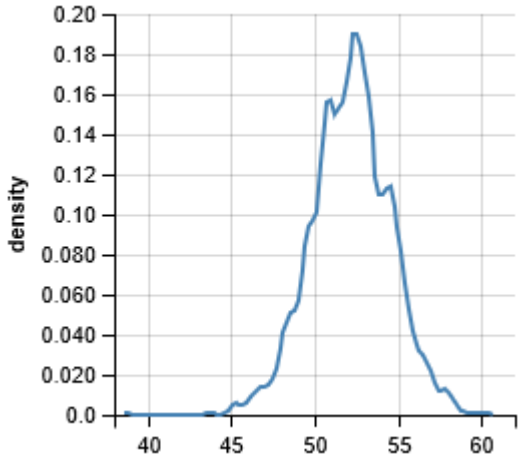

Figure 24: Task-1 Time Estimation of Tank

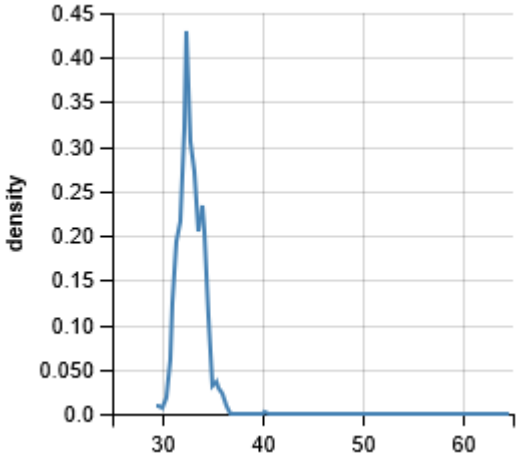

Figure 25: Task-4 Time Estimation of Tank

The task that was expected to take the longest time was the 5th task according to the Cogulator data, while it was the 1st task according to our model. The only task that does not receive any verbal instructions from the commander is the 1st task, and there is no time pressure on the user, he/she just performs the steps on instruction page sequentially. Other tasks, including the 5th task, take orders from the commander. In addition, for a value entered differently in 5th task, the system gives an error and asks the user to enter the new value by performing simple arithmetical calculation. One explanation for higher posteriors for the 1st task can be the absence of commander and time pressure. The posterior probability distribution of 5th task is presented in Figure 26.

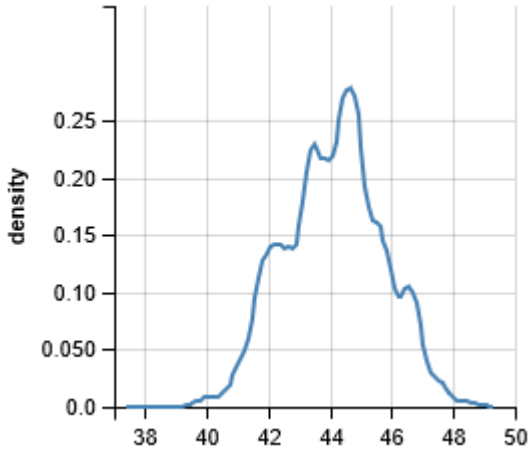

Figure 26: Task-5 Time Estimation of Tank

The largest change between the priors and posteriors were for the 4th task with 41.69 percent, and the lowest change was for the 2nd task with 2.77 percent.

Table 4

Task Completion Time Order in Tank Interface

Rank	Cogulator	Model
1	Task5	Task1
2	Task4	Task2
3	Task3	Task5
4	Task2	Task3
5	Task1	Task4

Table 4 ranks the execution time predictions of the Cogulator and Model in decreasing order. The orders are completely in this case. While Task 1 requires the shortest time in Cogulator estimates, it requires the longest time in the posteriors of the Bayesian model. Figures 24 – 28 shows the posterior distributions for all tasks.

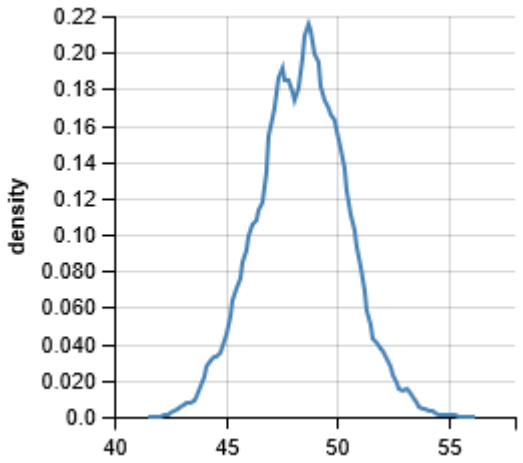

Figure 27: Task-2 Time Estimation of Tank

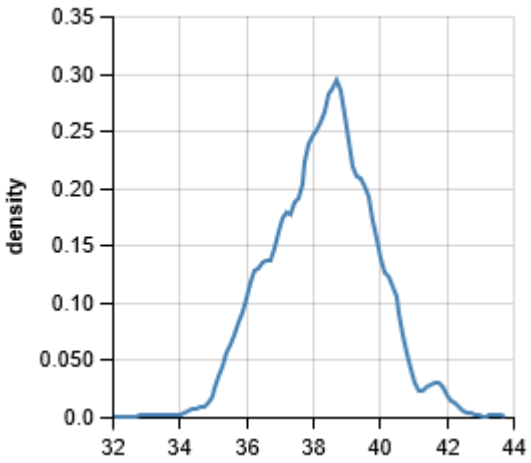

Figure 28: Task-3 Time Estimation of Tank

For the TCMS interface, prior time estimation of Cogulator and posterior time estimation of our model for each task are presented in Figure 29.

Figure 29: Prior and Posterior Execution Time of TCMS Interface Tasks

For our TCMS interface, 1st and 2nd tasks should be performed the fastest in this interface according to the Cogulator. Similarly, the Bayesian model also predicts 2nd task as the least time demanding. The 1st and 2nd tasks contain identical steps, so Cogulator estimates the same value. It is easier to follow the traces in the 2nd tasks than the 1st task as described in Section 4.1.1. While Cogulator cannot account for this difference, the Bayesian model could updated the Cogulator estimates based on the user data. Posterior distribution of the 1st and 2nd tasks are presented in Figure 30 and 31. Note that, Cogulator seem to underestimate execution time estimates in all tasks. Because the traces are randomly produced and it is not possible to predict which one will definitely enter the critical region. So the trace followed by user for a certain period of time and other traces' entering moment to the critical region could not be modeled clearly in the Cogulator.

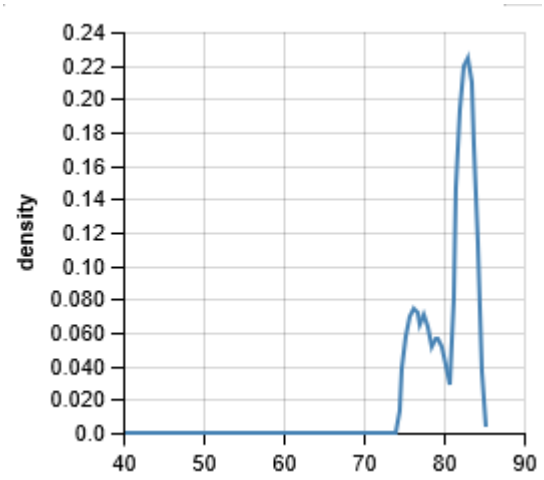

Figure 30: Task-1 Time Estimation of TCMS

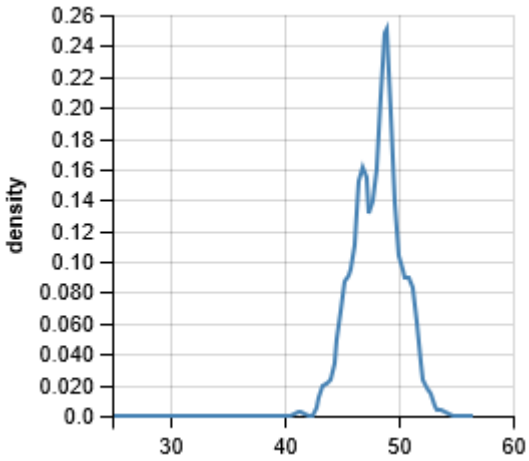

Figure 31: Task-2 Time Estimation of TCMS

The 5th task was expected to take the longest time both according to the Cogulator and our model. This task has more interaction steps than the other tasks. Figure 32 shows the posterior distribution of this task.

The 1st task has the highest difference between its prior and posterior with a 175.97 percent change, and the 5th task has the lowest with a 16.82 percent change.

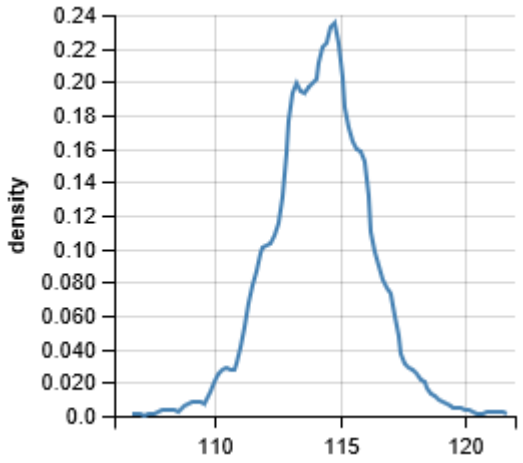

Figure 32: Task-5 Time Estimation of TCMS

Table 5

Task Completion Time Order in TCMS Interface

Rank	Cogulator	Model
1	Task5	Task5
2	Task3	Task1
3	Task4	Task3
4	Task1, Task2	Task4
5		Task2

The order of all tasks in the TCMS interface is given in Table 5 in terms of execution time. The tasks with longest time requirements are same according to the Cogulator and our model. Moreover, 2nd task requires shortest time to be performed for both Cogulator and our model. The posterior distribution of 3rd and 4th tasks are given in Figure 33 and 34.

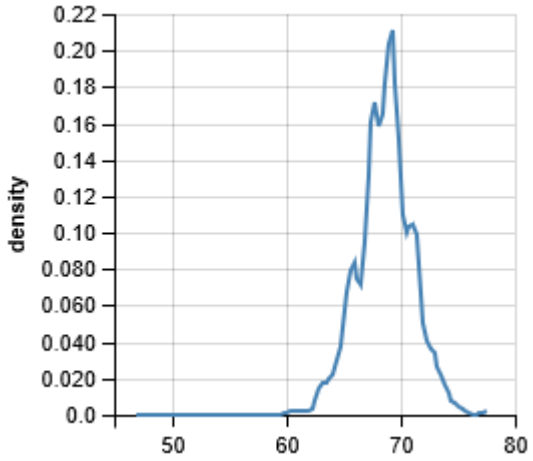

Figure 33: Task-3 Time Estimation of TCMS

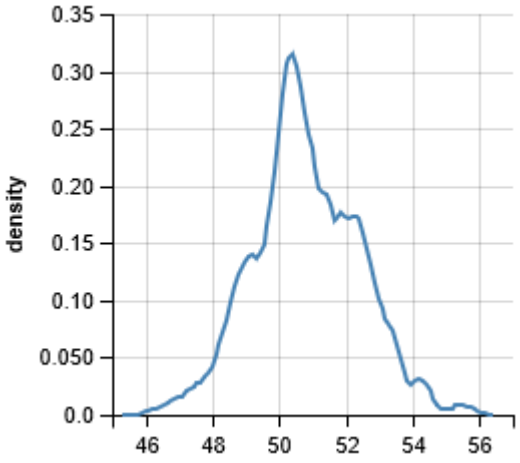

Figure 34: Task-4 Time Estimation of TCMS

4.2.2. Analyses of Users' Task Performance

We analyzed each user's task performance as relative to the average by using the *userSkill* variable in Type 2 Execution Time Model. Figure 35 shows the posterior *userSkill* for each user.

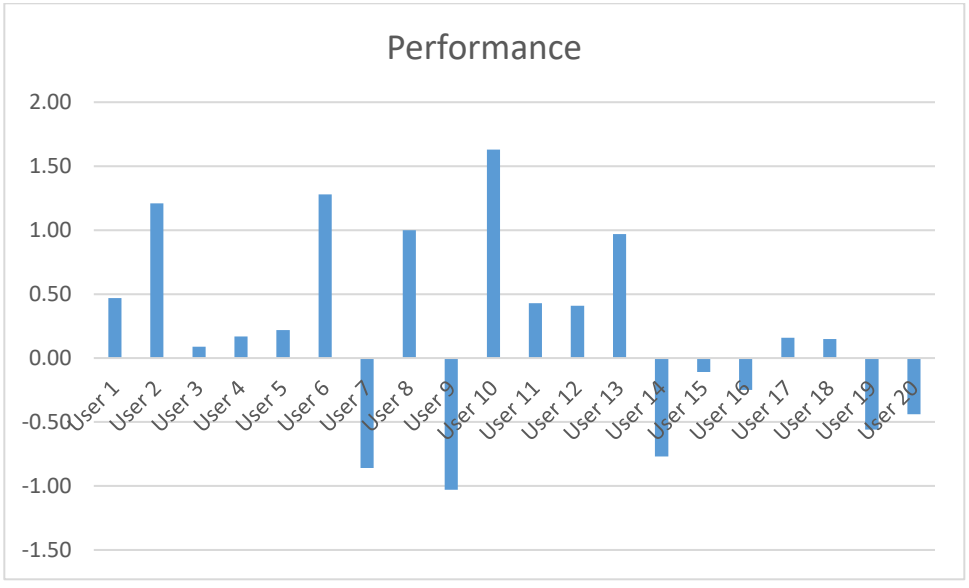

Figure 35: Task Performance of Users

According to these data; 13 out of 20 users need more performance than average, and 7 of them can complete tasks with less than average performance. User 10 (Figure 36) is at the

top of the average performance requirement, while user 9 (Figure 37) is at the bottom. This means that user 10 has the worst performance, while user 9 has the best performance.

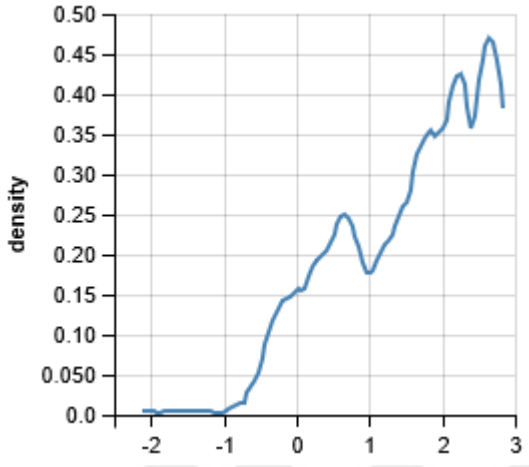

Figure 36: User-10 Performance

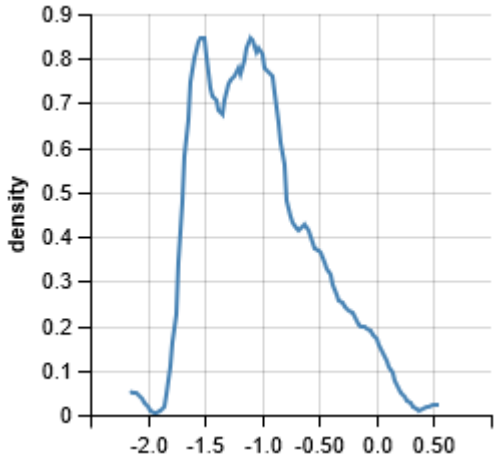

Figure 37: User-9 Performance

According to this data, three users with best performance are users 7, 9 and 14, and three users with the lowest performance are users 2, 6 and 10.

4.2.3. Analyses of Model's Predictive Performance

We also analyzed the predictive performance of the Type 2 Execution Time model by dividing the data into training and test sets with 80% to 20% ratio. We estimated the posterior completion time of each user for each task based on the training set. We then compared the prior time data of Cogulator and the posterior time data of model with real time. Figure 38 shows the true execution times, coagulator predictions (prior time) and the Bayesian model predictions (posterior time) for each experiment in the test sett. The mean absolute error is 18.9 for Cogulator, it is 9.0 for our model.

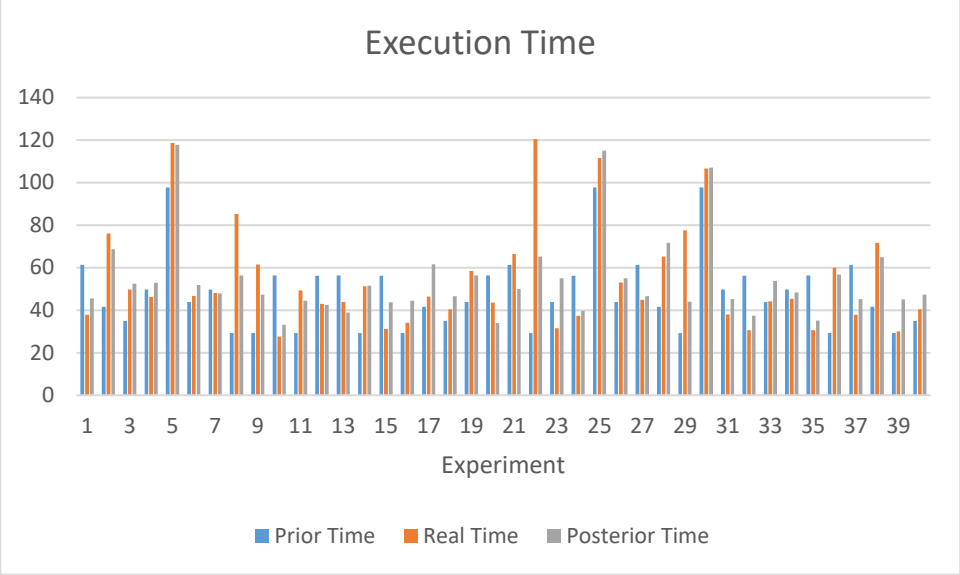

Figure 38: Prior, Real and Posterior Time

4.3. Summary of Results

According to the results of the analysis; a task's workload varies considerably in relation to the overall simplicity and complexity of the interface. Cogulator data produced priors in parallel order with the model in a simple interface where there are defined items and data on the screen, but since the screen complexity is high, it cannot evaluate the workload factors in the interface where random traces and tasks can occur. So it produced priors in different orders according to the model for this interface. So Cogulator workload data is more reliable when there is no randomness and high complexity in interface.

Analysis showed that the complexity of the interface affects not only the cognitive load requirement of the task but also the completion time of the task. If there is high complexity, user needs more time to find particular item and take action on interface. In addition, we observed that the task completion time decreased in the tasks where the commands came from the commander. So, prior and posterior estimation of execution time are not in the same order even for the simple interface. Task with the lowest workload and shortest time requirement was performed more slowly than others by users which has no command from commander. When commander gave orders, users performed more fast in general. This situation can be explained by the fact that the user feels both stress and time pressure when the commander gives an order.

When we analyzed user's task skills based on workload and their performance based on execution time, there is no relation between them. In other words, if user has the best task skill, it doesn't mean that this user will perform the task with shortest time. Similarly, if user performs tasks very fast, it doesn't guarantee that this user has very high task skill. When we analyze the highest and lowest task skills and performances of users, user 6 is

in the lowest three both for task skill and performance. But, user 10 is in the highest three for task skill while it is in the lowest three for performance.

Linear regression analysis identified that there is no significant relation between Bedford workload data of tasks and NASA-TLX overall workload data of interface.

According to the analysis results of models' predictive performance, our models estimated at least two times better than the Cogulator.

CHAPTER 5

DISCUSSION AND FUTURE WORK

5.1. Discussion

This study proposed a Bayesian approach for workload and performance measurement by combining estimates from GOMS model with data from subjective workload assessments and actual interface use. This study showed that GOMS do not provide reliable predictions as it cannot account for the differences between users due to hardware differences or many other parameters that can affect the results. In addition, general complexity of user interface and dynamic variables are not taken into account in GOMS models if they are not directly relational with designed tasks. But they play crucial role on cognitive load and performance. Moreover, cognitive ability and performance vary from person to person. Additionally, subjective workload measurements may be crucial to develop suitable user interfaces for the target audience, but developing an interface suitable for everyone is challenging in HCI. We proposed that Bayesian model can help to overcome these problems by combining multiple measurement and predictive methods.

Bayesian analysis provides a suitable way both to determine differences of computer based tasks of the user interface and subjective differences of the users. For instance, while our tank interface is usable and has simple tasks in general, the last task which has the highest workload according to the results forced experimenters. Because they have to keep and process a lot of information from the commander, so their working memory load has increased. For this reason, solutions such as sending some information to the sub-system in writing and displaying it on the screen instead of transmitting some information verbally by the commander, or automatizing the sent data, if possible, can be suggested. This improves the usability of the interface in general.

Two tasks of our torpedo counter measure system completely have same scenarios. Only difference is that in one scenario there are only relevant traces on the screen, while in the other scenario there are many different traces. Results showed that, workload of the task increases and user performance decreases in parallel with screen complexity. Information richness makes it difficult to see and detect critical traces in the threat class. For this, it can be suggested that the radar is technically improved to not to produce false traces. In terms of user experience and interface design; an optional filtering capability can be offered to the user so that unclassified and non-critical traces are not displayed in the user

interface. Non-critical and relatively unimportant traces can be indicated with less conspicuous colors and displayed graphically smaller than critical traces.

The outputs of personal differences computed our Bayesian model also provides useful information regarding the interfaces. Best three users based on performance analysis are system engineers who work on real systems in the field and have deep knowledge of the systems in general. Three users with the lowest performance are all working in design teams as industrial designer and mechanical design engineer. Estimation of personal differences by the Bayesian model provides a useful way to better understand the challenges and advantages encountered by the different groups of potential users.

5.2. Limitations and Future Studies

The first limitation of this study was the limited sample size of the participants and interfaces used in the experiments. The generalizability of the models can be further assessed with more samples on different user interfaces. Moreover, the experiments were carried out in a laboratory environment with previously prepared scenarios. Real time use of the models on interfaces that are in service can provide further evidence about their performance.

In this study, Bayesian models were built at the task level. More detailed models can be built by representing each sub-task of these tasks and adding more layers to the BN. Increased complexity of these models will require collecting more data from the users at sub-task level.

Our Bayesian model is based on predictive cognitive model and subjective measurement techniques. Other measurement techniques for workload and performance measurement include physiological and behavioral methods. Data collected by these techniques such as eye tracking, observing heart rate, monitoring brain activity or mouse tracking can be included in future studies to provide a more comprehensive Bayesian model of cognitive load and performance measurement.

REFERENCES

Akgun, M., Akilli, G. K., & Cagiltay, K. (2011). Bringing affect to human computer interaction. In *Affective Computing and Interaction: Psychological, Cognitive and Neuroscientific Perspectives* (pp. 308-324). IGI Global.

Baştürk, Ö., n.d. *Ders06_Monte_Carlo_Yontemleri_Bayesian_Istatistige_Giris*. [online] Ozgur.astrotux.org. Available at: <http://ozgur.astrotux.org/ast416/Ders_06/Ders06_Monte_Carlo_Yontemleri_Bayesian_Istatistige_Giris.html> [Accessed 12 August 2021].

Bernardo, J. M., Bayarri, M. J., Berger, J. O., Dawid, A. P., & Heckerman, D. (2011). *Bayesian statistics 9* (Vol. 9). Oxford University Press.

Besson, P., Bourdin, C., Bringoux, L., Dousset, E., Maïano, C., Marqueste, T., ... & Vercher, J. L. (2013). Effectiveness of physiological and psychological features to estimate helicopter pilots' workload: A Bayesian network approach. *IEEE Transactions on Intelligent Transportation Systems*, *14*(4), 1872-1881.

Brookhuis, K. A., & De Waard, D. (2010). Monitoring drivers' mental workload in driving simulators using physiological measures. *Accident Analysis & Prevention, 42*(3), 898-903.

Card, S. K., Moran, T. P., & Newell, A. (1983). The Psychology of. *Human-Computer Interaction*, 1-43.

Casner, S. M., & Gore, B. F. (2010). Measuring and evaluating workload: A primer. *NASA Technical Memorandum*, 216395, 2010.

Castelletti, F. (2020). Bayesian model selection of Gaussian directed acyclic graph structures. *International Statistical Review, 88*(3), 752-775.

Chen, S. H., & Pollino, C. A. (2012). Good practice in Bayesian network modelling. *Environmental Modelling & Software*, *37*, 134-145.

Nasa.gov. 2020. *Cognitive Workload*. [online] Available at: <https://www.nasa.gov/sites/default/files/atoms/files/cognitive_workload_technical_brief_ochmo_06232020.pdf> [Accessed 2 August 2021].

Conati, C., & VanLehn, K. (2001, January). Providing adaptive support to the understanding of instructional material. In *Proceedings of the 6th international conference on Intelligent user interfaces* (pp. 41-47).

Dudley, J. J., Jacques, J. T., & Kristensson, P. O. (2019, May). Crowdsourcing interface feature design with Bayesian optimization. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (pp. 1-12).

Elkin-Frankston, S., Bracken, B. K., Irvin, S., & Jenkins, M. (2017). Are behavioral measures useful for detecting cognitive workload during human-computer interaction?. In *Advances in The Human Side of Service Engineering* (pp. 127-137). Springer, Cham.

Fischer, G. (2001). User modeling in human–computer interaction. *User modeling and user-adapted interaction*, 11(1), 65-86.

Gao, Z., & Wang, S. (2015, June). Emotion recognition from EEG signals using hierarchical Bayesian network with privileged information. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval* (pp. 579-582).

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (1995). *Bayesian data analysis*. Chapman and Hall/CRC.

Gokcay, D., & Yildirim, G. (2011). *Affective computing and interaction: Psychological, cognitive, and neuroscientific perspectives*. IGI Global (701 E. Chocolate Avenue, Hershey, Pennsylvania, 17033, USA).

Goodman, N. D., & Tenenbaum, J. B. The ProbMods Contributors (2016). Probabilistic Models of Cognition.

Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology* (Vol. 52, pp. 139-183). North-Holland.

Hart, S. G. (2006, October). NASA-task load index (NASA-TLX); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting* (Vol. 50, No. 9, pp. 904-908). Sage CA: Los Angeles, CA: Sage publications.

Hochstein, L. (2002). Goms. *Theories in Computer Human Interaction, University of Maryland, College Park, MD, USA*.

Huang, Y. P., Chang, H. C., & Lin, C. C. (2011, June). Systematic design of environmental monitoring interface by Bayesian classification. In *Proceedings 2011 International Conference on System Science and Engineering* (pp. 43-48). IEEE.

John, B. E., & Kieras, D. E. (1996). The GOMS family of user interface analysis techniques: Comparison and contrast. *ACM Transactions on Computer-Human Interaction* (TOCHI), 3(4), 320-351.

Jorritsma, W., Haga, P. J., Cnossen, F., Dierckx, R. A., Oudkerk, M., & van Ooijen, P. M. (2015). Predicting human performance differences on multiple interface alternatives: KLM, GOMS and CogTool are unreliable. *Procedia Manufacturing*, 3, 3725-3731.

Kaptelinin, V., Nardi, B., Bødker, S., Carroll, J., Hollan, J., Hutchins, E., & Winograd, T. (2003, April). Post-cognitivist HCI: second-wave theories. In *CHI'03 extended abstracts on Human factors in computing systems* (pp. 692-693).

Kieras, D. E., Wood, S. D., Abotel, K., & Hornof, A. (1995, December). GLEAN: A computer-based tool for rapid GOMS model usability evaluation of user interface designs. In *Proceedings of the 8th annual ACM symposium on User interface and software technology* (pp. 91-100).

Kieras, D. E. (1999). A guide to GOMS model usability evaluation using GOMSL and GLEAN3. *University of Michigan, 313*.

Klingner, J. (2010). *Measuring cognitive load during visual tasks by combining pupillometry and eye tracking*. Stanford University.

Ko, K. E., & Sim, K. B. (2009). Development of facial expression recognition system based on bayesian network using FACS and AAM. *Journal of Korean Institute of Intelligent Systems*, 19(4), 562-567.

Kovesdi, C. R., & Joe, J. C. (2019, November). Exploring The Use of Cognitive Models for Nuclear Power Plant Human-System Interface Evaluation. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 63, No. 1, pp. 2190-2194). Sage CA: Los Angeles, CA: SAGE Publications.

Lee, M. D., & Wagenmakers, E. J. (2014). *Bayesian cognitive modeling: A practical course*. Cambridge university press.

Liaghati, C., Mazzuchi, T., & Sarkani, S. (2020). A method for the inclusion of human factors in system design via use case definition. *Human-Intelligent Systems Integration*, 2(1), 45-56.

Lu, P., Huang, X., Zhu, X., & Wang, Y. (2005, June). Head gesture recognition based on bayesian network. In *Iberian Conference on Pattern Recognition and Image Analysis* (pp. 492-499). Springer, Berlin, Heidelberg.

Luo, R., Wang, Y., Weng, Y., Paul, V., Brudnak, M. J., Jayakumar, P., ... & Yang, X. J. (2019, November). Toward real-time assessment of workload: a Bayesian inference approach. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 63, No. 1, pp. 196-200). Sage CA: Los Angeles, CA: SAGE Publications.

MacKenzie, I. S. (2012). Human-computer interaction: An empirical research perspective.

Miller, S. (2001). Workload measures. *National Advanced Driving Simulator. Iowa City, United States.*

Mihaljević, B., Bielza, C., & Larrañaga, P. (2021). Bayesian networks for interpretable machine learning and optimization. *Neurocomputing*.

Moré, A. G. (2014). A Quantitative Evaluation of Pilot-in-the-Loop Flying Tasks Using Power Frequency and NASA TLX Workload Assessment.

Nguyen, L., & Do, P. (2009, May). Combination of Bayesian network and overlay model in user modeling. In *International Conference on Computational Science* (pp. 5-14). Springer, Berlin, Heidelberg.

N. D. Goodman and A. Stuhlmüller (electronic). The Design and Implementation of Probabilistic Programming Languages. Retrieved 2021-8-23 from `http://dippl.org`.

Patel, V. L., & Kushniruk, A. W. (1998). Interface design for health care environments: the role of cognitive science. In *Proceedings of the AMIA Symposium* (p. 29). American Medical Informatics Association.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann. https://doi.org/10.1016/C2009-0-27609-4

Ramkumar, A., Stappers, P. J., Niessen, W. J., Adebahr, S., Schimek-Jasch, T., Nestle, U., & Song, Y. (2017). Using GOMS and NASA-TLX to evaluate human–computer interaction process in interactive segmentation. *International Journal of Human–Computer Interaction*, 33(2), 123-134.

Rim, R., Amin, M. M., & Adel, M. (2013, December). Bayesian networks for user modeling: Predicting the user's preferences. In *13th International Conference on Hybrid Intelligent Systems (HIS 2013)* (pp. 144-148). IEEE.

Riva, G., Vatalaro, F., & Davide, F. (Eds.). (2005). *Ambient intelligence: the evolution of technology, communication and cognition towards the future of human-computer interaction* (Vol. 6). IOS press.

Roscoe, A. H. (1984). *Assessing pilot workload in flight*. ROYAL AIRCRAFT ESTABLISHMENT BEDFORD (UNITED KINGDOM) BEDFORD United Kingdom.

Ruokangas, C. C., & Mengshoel, O. J. (2003, January). Information filtering using bayesian networks: *effective user interfaces for aviation weather data. In Proceedings of the 8th international conference on Intelligent user interfaces* (pp. 280-283).

Rozado, D., & Dunser, A. (2015). Combining EEG with pupillometry to improve cognitive workload detection. *Computer, 48*(10), 18-25

Rubio, S., Díaz, E., Martín, J., & Puente, J. M. (2004). Evaluation of subjective mental workload: A comparison of SWAT, NASA-TLX, and workload profile methods. *Applied psychology, 53*(1), 61-86.

Shneiderman, B., & Plaisant, C. (2010). *Designing the user interface: Strategies for effective human-computer interaction.* Pearson Education India.

Scholtz, J. (2004). Usability evaluation. *National Institute of Standards and Technology, 1*.

Sebe, N., Cohen, I., Huang, T. S., & Gevers, T. (2005, July). Human-computer interaction: a Bayesian network approach. In *International Symposium on Signals, Circuits and Systems, 2005. ISSCS 2005*. (Vol. 1, pp. 343-346). IEEE.

Song, I. J., & Cho, S. B. (2013). Bayesian and behavior networks for context-adaptive user interface in a ubiquitous home environment. *Expert Systems with Applications, 40*(5), 1827-1838.

Stephenson, T. A. (2000). *An introduction to Bayesian network theory and usage* (No. REP_WORK). IDIAP.

Sweller, J. (2018). Measuring cognitive load. *Perspectives on medical education, 7*(1), 1-2.

Yuan, H., Li, S., & Rusconi, P. (2020). *Cognitive Modeling for Automated Human Performance Evaluation at Scale*. Springer Nature.

Zhang, Y., Zheng, H., Duan, Y., Meng, L., & Zhang, L. (2015, June). An integrated approach to subjective measuring commercial aviation pilot workload. In *2015 IEEE 10th Conference on Industrial Electronics and Applications* (ICIEA) (pp. 1093-1098). IEEE.

Zheng, Y., & Jie, Y. (2019, July). Study of NASA-TLX and Eye Blink Rates Both in Flight Simulator and Flight Test. *In International Conference on Human-Computer Interaction* (pp. 353-360). Springer, Cham.

APPENDICES

APPENDIX A

UYGULAMALI ETİK ARAŞTIRMA MERKEZİ
APPLIED ETHICS RESEARCH CENTER

DUMLUPINAR BULVARI 06800
ÇANKAYA ANKARA/TURKEY
T: +90 312 210 22 91
F: +90 312 210 79 59
ueam@metu.edu.tr
www.ueam.metu.edu.tr

ORTA DOĞU TEKNİK ÜNİVERSİTESİ
MIDDLE EAST TECHNICAL UNIVERSITY

Sayı: 28620816 /

26 Temmuz 2021

Konu : Değerlendirme Sonucu

Gönderen: ODTÜ İnsan Araştırmaları Etik Kurulu (İAEK)

İlgi : İnsan Araştırmaları Etik Kurulu Başvurusu

Sayın Dr. Barbaros YET

Danışmanlığını yürüttüğünüz Aysun SAYDAM'ın "Arayüzle Etkileşime Giren Kullanıcının Bilişsel Yükünün Bayes Ağına Dayalı Tahmini" başlıklı araştırması İnsan Araştırmaları Etik Kurulu tarafından uygun görülmüş ve **297-ODTU-2021** protokol numarası ile onaylanmıştır.

Saygılarımızla bilgilerinize sunarız.

Prof.Dr. Mine MISIRLISOY
İAEK Başkan

APPENDIX B

Cogulator Model of 1st Task in Tank Interface

operator	label	line_number	resource	thread	operator_time	step_start_time	step_end_time
look	at the X button	0	see	base	550	0	550
point	to the X button	1	hands	base	950	550	1500
click	on the X button	2	hands	base	320	1500	1820
look	at the Y button	3	see	base	550	1820	2370
point	to the Y button	4	hands	base	950	2370	3320
click	on the Y button	5	hands	base	320	3320	3640
look	at the Tamam button	6	see	base	550	3640	4190
point	to the Tamam button	7	hands	base	950	4190	5140
click	on the Tamam button	8	hands	base	320	5140	5460
look	at the Z button	9	see	base	550	5460	6010
point	to the Z button	10	hands	base	950	6010	6960
click	on the Z button	11	hands	base	320	6960	7280
look	at the M button	12	see	base	550	7280	7830
point	to the M button	13	hands	base	950	7830	8780
click	on the M button	14	hands	base	320	8780	9100
look	at the MS button	15	see	base	550	9100	9650
point	to the MS button	16	hands	base	950	9650	10600
click	on the MS button	17	hands	base	320	10600	10920
look	at the KT button	18	see	base	550	10920	11470
point	to the KT button	19	hands	base	950	11470	12420
click	on the KT button	20	hands	base	320	12420	12740
look	at the TS button	21	see	base	550	12740	13290
point	to the TS button	22	hands	base	950	13290	14240
click	on the TS button	23	hands	base	320	14240	14560
look	at the T button	24	see	base	550	14560	15110
point	to the T button	25	hands	base	950	15110	16060
click	on the T button	26	hands	base	320	16060	16380
look	at the MR button	27	see	base	550	16380	16930
look	at the H button	28	see	base	550	16930	17480
point	to the H button	29	hands	base	950	17480	18430
click	on the H button	30	hands	base	320	18430	18750
look	at the A image	31	see	base	550	18750	19300
point	to the R button	32	hands	base	950	19300	20250
click	on the Rbutton	33	hands	base	320	20250	20570
look	at the A image	34	see	base	550	20570	21120
point	to the R button	35	hands	base	950	21120	22070
click	on the R button	36	hands	base	320	22070	22390
look	at the A image	37	see	base	550	22390	22940
point	to the A button	38	hands	base	950	22940	23890
click	on the R button	39	hands	base	320	23890	24210
look	at the A image	40	see	base	550	24210	24760
point	to the B button	41	hands	base	950	24760	25710
click	on the B button	42	hands	base	320	25710	26030
look	at the B image	43	see	base	550	26030	26580
point	to the B button	44	hands	base	950	26580	27530
click	on the B button	45	hands	base	320	27530	27850
look	at the A image	46	see	base	550	27850	28400
point	to the B button	47	hands	base	950	28400	29350
click	on the B button	48	hands	base	320	29350	29670
look	at the IA textbox	49	see	base	550	29670	30220
point	to the IA textbox	50	hands	base	950	30220	31170
click	on the IA textbox	51	hands	base	320	31170	31490

think	of text to type	52	cognitive	base	1250	31490	32740
hands	to keyboard	53	hands	base	450	32740	33190
type	125	54	hands	base	840	33190	34030
look	at the Y textbox	55	see	base	550	34030	34580
hands	to mouse	56	hands	base	450	34580	35030
point	to the Y textbox	57	hands	base	950	35030	35980
click	on the Y textbox	58	hands	base	320	35980	36300
think	of text to type	59	cognitive	base	1250	36300	37550
hands	to keyboard	60	hands	base	450	37550	38000
type	243	61	hands	base	840	38000	38840
look	at the B value	62	see	base	550	38840	39390
hands	to mouse	63	hands	base	450	39390	39840
point	to the P button	64	hands	base	950	39840	40790
click	on the Pbutton	65	hands	base	320	40790	41110
look	at the B value	66	see	base	550	41110	41660
click	on the P button	67	hands	base	320	41660	41980
look	at the A button	68	see	base	550	41980	42530
point	to the A button	69	hands	base	950	42530	43480
click	on the A button	70	hands	base	320	43480	43800

APPENDIX C

Cogulator Model of 1st Task in TCMS Interface

operator	label	line_number	resource	thread	operator_time		step_start_time	step_end_time	
look	at the Menu button	0	see	base	550	0	550		
point	to the Menu button	1	hands	base	950	550	1500		
click	on the Menu button	2	hands	base	320	1500	1820		
look	at the CIT button	3	see	base	550	1820	2370		
point	to the CIT button	4	hands	base	950	2370	3320		
click	on the CIT button	5	hands	base	320	3320	3640		
look	at the TAMAM button	6	see	base	550	3640	4190		
point	to the TAMAM button	7	hands	base	950	4190	5140		
click	on the TAMAM button	8	hands	base	320	5140	5460		
look	at the İz_Listesi button	9	see	base	550	5460	6010		
point	to the İz_Listesi button	10	hands	base	950	6010	6960		
click	on the İz_Listesi button	11	hands	base	320	6960	7280		
hear	<T26> yı takip et, kiritik bölgeye girerse imha et.Aynı zamanda						…….		
	13	hear	base	7200	7280	14480			
attend	to U42	15	cognitive	base	50	14480	14530		
hands	to keyboard	16	hands	base	450	14530	14980		
type	U42	17	hands	base	840	14980	15820		
attend	to U41	19	cognitive	base	50	15820	15870		
type	U41	20	hands	base	840	15870	16710		
attend	to T26	23	cognitive	0	50	15870	15920		
initiate	Eye movement to T26	24	cognitive	base	50	16710	16760		
hands	to mouse	27	hands	0	450	16710	17160		
saccade	to T26	25	see	base	30	16760	16790		
look	at target	28	see	0	550	17160	17710		
look	at T26	26	see	base	100	16790	16890		
point	to T26	29	hands	0	950	17710	18660		
look	at the T26 button	37	see	base	550	17710	18260		
cognitive_processor		Verify Cursor is over T26	30	cognitive	0	70	18660	18730	
look	at target	38	see	base	550	18260	18810		
attend	to Karistir	31	cognitive	0	50	18730	18780		
point	to the T26 button	39	hands	base	950	18810	19760		
initiate	Click Karistir	32	cognitive	0	50	18780	18830		
click	on the T26 button	40	hands	base	320	19760	20080		
look	at target	33	see	0	550	18830	19380		
look	at the Derinlik	41	see	base	550	20080	20630		
point	to target	34	hands	0	950	20080	21030		
look	at <4580>	42	see	base	550	20630	21180		
click		35	hands	0	90	21030	21120		
look	at the Lancer button	44	see	base	550	21180	21730		
point	to the Lancer button	45	hands	base	950	21730	22680		
click	on the Lancer button	46	hands	base	320	22680	23000		
hear	Sancak-4 karistiricisini at.	49	hear	base	1200	23000	24200		
look	at the Sancak-4 button	51	see	base	550	24200	24750		
point	to the Sancak-4 button	52	hands	base	950	24750	25700		
click	on the Sancak-4button	53	hands	base	320	25700	26020		

hands	to keyboard	55	hands	base	450	26020	26470		
type	<4580>	56	hands	base	1120	26470	27590		
look	at the Tamam button		57	see	base	550	27590	28140	
hands	to mouse	58	hands	base	450	28140	28590		
point	to the Tamam button		59	hands	base	950	28590	29540	
click	on the Tamam button		60	hands	base	320	29540	29860	
read	sancak-4 gönderilecek, emin misiniz?		62	see	base	1040	29860	30900	
look	at the Tamam button		63	see	base	550	30900	31450	
point	to the Tamam button		64	hands	base	950	31450	32400	
click	on the Tamam button		65	hands	base	320	32400	32720	