On User Modeling

Kelly Storm
Black Box Operations

Dec. 1st, 2009

1 Abstract

Web characterization and user modeling in the field of computer science has grown to include attributes as simple as geographical location and as complex as patterns of online behavior. While these user models produce useful results, their predictive power is limited by not taking the users' initiatives, reasoning, and goals are rarely taken into consideration. In the field of cognitive psychology, user models are created based on the users' responses to questionnaires and actions within a system. The user models created by methodologies such as ACT-R and GOMS have been applied to accurately predict the effectiveness of a machine or graphical user interface. While these systems incorporate the users' goals into the user model, the majority of the research has involved analyzing simple menus and basic user interfaces. We feel that we can create more effective user models by combining the user models found within both realms of science. To do so, we propose a study that would extend ACT-R to advanced web applications and provide better models for predicting user behavior. We also discuss potential future work if the assumptions within this paper are solidified with the results of the proposed study.

2 Introduction

Traditionally, web characterization and user modeling in the field of computer science has involved creating models that include attributes such as path traversal behavior, visit frequency, staying time, geographical location, ISP, and ranked or marked page requests. The user models can be created from parsing and sessionizing web server access logs, or audit trails, but their predictive power is limited to statistical approaches such as regression analysis. User models are also typically confined to information that has been annotated, or created, based on the evaluators' and researchers' assumptions. For example, clustering algorithms are often created and trained based on data sets that have been annotated by researchers who assume they understand the users' intentions and goals [1]. While we can understand cognitive and metacognitive processes by analyzing web server access logs, or audit trails, the predictive models created within the field of computer science generate results and create reports that ignore the users' cognitive and metacognitive processes [2]. However, this is not due to the lack of initiative to incorporate the users' needs into a user model, but rather lacking the knowledge of how to properly classify users based on their actions and their goals.

In the realm of cognitive psychology, cognitive user models are created based on user-completed questionnaires or by observing user actions, and making inferences based on stored knowledge [2]. Predictive cognitive user models, such as the task-performance model created by GOMS and the production-goal model created by ACT-R, are used to predict user behavior based on stored knowledge [2, 3]. Predictive user models can be used to make predictions about how effectively the users can complete specific tasks within a machine or graphical user interface. However, most of the research conducted in the field of cognitive psychology that apply these predictive user models is contained to menu interaction [3, 4, 5, 6]. Nonetheless, research using predictive cognitive models has produced accurate, attractive results supporting the idea that models can realistically mimic and predict human user behavior.

Computer applications such as desktop and web applications have become an integral part of daily life. The pervasive nature of the personal computer has almost every industry in the world using applications to assist in daily business procedures where information is created, read, updated, deleted, and queried against. The applications are also becoming increasingly complex. It is clear that while the fields of computer science and cognitive psychology are capable of creating effective predictive user models separate of one another, merging the two together may eliminate some of the weaknesses existing in the current models. With significant research involving efficient data mining and characterizing web users based on their behavior, our understanding of online activity exceeds the analysis of simple contextual menus [7, 8, 1]. We feel that we can create a more effective predictive cognitive model by combining the idea of cognitive models with the information derived from web usage mining and behavior characterization techniques found within computer science.

To test our ideas we propose a study that will combine the ACT-R technique with a common web user model, a Customer Behavior Model Graphs (CBMG), a data extraction technique known as access log sessionizing, and explicit exposure to key productions most commonly referred to as tutorials. We also propose extending ACT-R to work with dynamic web applications involving complex actions outside the realm of menu traversal and interaction.

3 Background

3.1 Web Log Mining

A web server's access log is a program or physical machine that delivers content to web users requesting information from a third party, such as Amazon or Google. Whenever a web user requests information from a web server, the request is logged into a file called a web server access log. These web server access logs contain a wealth of information, and several effective systems have been created to mine web access logs for useful patterns of traversal through a website, demographic information, popularity of content, visit duration, and probability of revisit [9, 10, 8, 11, 6]. A prime example of the information that can be derived from a web access log is that of Johnson's work [12]. Johnson mined the CDC's web access log for user searches of pages containing content related to influenza and was able to determine the outbreak of influenza up to twenty weeks before the outbreak became severe.

Based on our observations, these logs are referred to as "access logs" in the realm of computer science, and "audit trails" in the realm of cognitive psychology. We will refer to these logs as "access logs" for the duration of the paper.

A session is defined as a sequence of requests delimited by an interval of time, typically thirty minutes, made by an individual user. Sessionizing an access log involves parsing a web server access log into sessions where each session belongs to a specific user and every session is unique. An episode is a sequence of requests found within any given session.

3.2 Web User Characterization

Web users can be characterized by a variety of attributes. These attributes include, but are not limited to, path traversal behavior, page and visit frequency, page temporal information, and transaction information.

Workload generation is a common form of web user characterization. This type of performance evaluation technique predicts the performance to which quality of service requirements are met. By modeling the behavior of users using a web application we can study expected traffic characteristics and the impact of the network on the services and applications by the user [13]. The effectiveness of workload generation depends on the accuracy of the model, and can thus sometimes generate incorrect predictions.

User profiling is another useful way to characterize web users. A profile is defined as a list of URLs and their corresponding frequencies of existence for a set of user sessions that have been clustered or typed [14]. Profiles can be used to determine the popularity of content by user type. For example, if a cluster of user sessions that have been typed as those created by advanced users has a high frequency of existence of a particular episode, we can assume that this area of the site is helpful to advanced users. We could also infer that other types of users have avoided this area of the site either intentionally due to its complexity or unintentionally due to is placement within the site. Profiling allows evaluators to understand the website's areas of interest for specific types of users.

A page's temporal information can also be used to characterize web users. Temporal information is derived from temporal web log mining, which takes the staying time of a user on each page into consideration [15]. Staying time is the amount of time elapsed between user requests and can be derived from the access log. Temporal information has previously been used to improve a web site's effectiveness by indicating potentially troublesome portions of the site based on a higher than average staying time for particular pages or types of pages. Temporal web log mining has also been used in conjunction with latent semantic indexing to cluster pages based on their content then compare average staying times for all pages within the group [16, 17]. By applying temporal web log mining to a dataset, evaluators can quickly determine potential problematic areas within the website based on the users' higher than average staying time and investigate accordingly.

Other methods of web user characterization include:

- Structural link analysis from user profiles within web sites such as LiveJournal and Facebook to determine the relevancy of links between profiles
- Path traversal pattern data mining to observe the statistically significant longest common subsequences of paths through a website
- Characterizing workloads with transaction oriented views to correlate specific systems operations to episodes which are linked to real world tasks such as generating financial reports
- User interface profiling which generates user profiles based on traversal patterns through a web site [18, 8, 7, 19]

3.3 Web User Models

Web user models are a way to characterize the overall behavior of a web user. Web user models can be informative or predictive, and they can also be used to generate type-characteristic data. An example of a common web user model is a Customer Behavior Model Graph, or CBMG. CBMGs show the transition frequencies of a user or group of users traversing from one page to the next for all pages found within the session set [20]. CBMGs are often accompanied by visualizations where the pages are nodes and the transitions are edges labeled with the transition frequency. CBMGs can be created for a single user or a group of users that have been clustered by characteristics such as ISP or geographical location. A CBMG is created first by tallying the transitions from one page to the next of a user or group of users. Table 1 shows the tallied transitions between five pages.

-	1	2	3	4	5
1	3	5	2	1	0
2	2	0	4	6	3
3	0	0	0	8	4
4	1	3	7	0	1
5	3	4	5	4	0

Table 1: Transition Counts for User Group

Next, a transition frequency matrix is created by computing the probability to transition from one page to the next of the users within the group. This computation consists of tallying the total number of transitions from one page to its respective out pages. For example, the first row of Table 2 shows the probability of the users to transition from page one to page one, page one to page two, page one to page three, and so on.

-	1	2	3	4	5
1	27.3%	45.5%	18.2%	9%	0%
2	13.4%	0%	26.6%	40%	20%
3	0%	0%	0%	66.7%	33.3%
4	8.4%	24.9%	58.3%	0%	8.4%
5	18.6%	25%	31.4%	25%	0%

Table 2: Transition Frequency Matrix

Finally, a CBMG is generated depicting the pages as nodes and the transition frequencies as edges. The CBMG shown below in Figure 1 is capable of providing a type of user model that allows researchers and evaluators to make predictions based on the type of user the CBMG represents. To un-

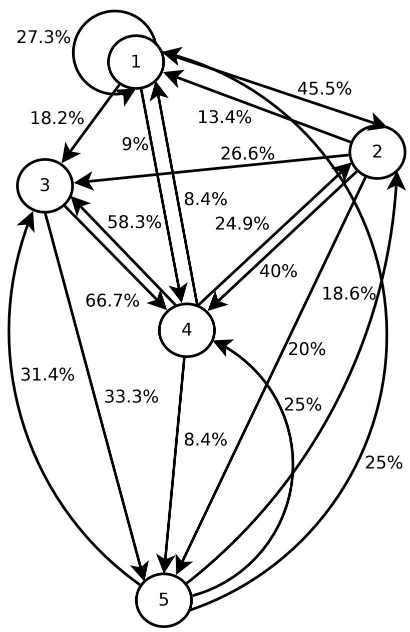

Figure 1: Customer Behavior Model Graph

derstand how this model can be used, assume the CBMG was created from sessions that had previously been annotated as those belonging to advanced users of a website. The model can answer specific questions, such as whether advanced users are more likely to go from page three to page four, without having to query the access log directly.

Clustering algorithms are often used to generate typed groups before creating a user model. Sometimes the clustering algorithms use methods such as latent semantic indexing, which identifies pages with similar content and groups user sessions based on this information [16]. Other times, clustering algorithms are used to comprehend web sites and web applications by analyzing longest common subsequences and evaluator specified paths to enhance the navigation structure of web sites based on user behavior [21, 16, 17].

Some work has been done to create models which comprehend and correctly predict the evolution of a user from one type to the next, such as novice to expert [22]. However, most of the work published regarding user models relies solely on basic characteristics, such as simple path traversal, transition frequency, staying time, and geographic location.

3.4 Cognitive User Models

Cognitive user models can be divided into two types, predictive and personalized. Models are typically developed for specific tasks such as menu traversal and then compared to human data to determine the validity of the model itself [2].

Developing a predictive user model involves performing task analysis to define the complexity of the task. The information provided by the task analysis, used within an architecture of cognition, can be used to make predictions about human performance, such as reaction time and error rate. An example of a predictive user model is the GOMS technique in which a task is broken down into goals, operators, methods, and selection rules. Using a GOMS analysis allows researchers to understand a task at the cognitive level. Given a task, we can understand the sub-goals involved with reaching the main goal, the operators used, and the choices made. Other models also exist, such as MHP, which annotates a GOMS analysis by attaching predicted execution times, EPIC, which bases its analysis on productions, SOAR, which can dynamically create new rules and productions as the task unfolds, and ACT-R, which integrates learning and forgetting into the user model [2]. In essence, a predictive user model is attempting to simulate human behavior by encoding long term memory, working memory, perception, cognition, and motor processing as tasks, productions, and goals. A production is an annotated set of cognitive operations used to reach a goal and will be used in later sections when we attempt to integrate user models.

Personalized user models take the same information used by predictive user models into consideration. However, a personalized user model takes user interactions into consideration to dynamically alter the task itself. personalized user models can be used to create adaptable and adaptive systems, the difference being that adaptable systems allow the user to configure parameters explicitly, and adaptive systems configure the system automatically based on a user's behavior. A personalized user model can thus be used to determine which elements should be configurable for an adaptable system and which elements should dynamically adapt for an adaptive system.

4 Related Work

For our research we focused on related work that applied cognitive user models to graphical user interfaces. By understanding how predictive cognitive user models are applied to user interfaces we feel we can better combine and apply the user models found in the field of computer science with those found in the field of cognitive psychology.

AMME, or Automatic Mental Model Evaluation, was developed to better understand user behavior in the field of cognitive ergonomics. AMME analyzed an action sequence and generated a description of the task dependent model of the user, a state transition matrix, and quantitative measures of the task solving process. AMME generated a Petri Net containing states and transitions where the states were elements in a menu and the transitions were actions made by the user [23]. One of the most significant contributions of AMME is that of depicting parts of a problem as states and user actions as transitions.

Ritter's work used a predictive model created with ACT-R to program a robot to perform actions within a simulation of real world events, the event being driving a car. The model, created from data generated by humans, was able to make realistic predictions on human behavior such as the degree to which fatigue effects driving performance [5]. Ritter's work helped expand the domain of ACT-R past understanding the productions and goals of solving simple mathematical equations.

Amant and Das have performed several model based evaluations on cell phone menus and keypads using predictive user models. Using their predictive cognitive model they were able to accurately predict how long specific actions would take for human users without having to manually test the devices. The results generated by these studies lead to a programmatic implementation of the predictive user model which can be used to test new cell phone interfaces without having to create the physical device or hold human user studies [24, 6]. The significance of Amant's and Das's work is that of directly applying a cognitive user model to a graphical user interface.

The most relevant related work is Lynch, Palmiter and Tilt's research behind creating a standard web site user model, called the Max Model. The Max Model is comprised of a set of characteristics including "cultural characteristics, psychological characteristics, training and experience characteristics, internet connections, system characteristics, and cognitive capabilities" [25]. Their research validated the use of a cognitive web user model, and incorporated website and web user characteristics into usability metrics. The Max Model included attributes such as the ability to scan the user interface for new elements, waning human persistence, and latency for detection, decision and selection. The significance of Lynch, Palmiter and Tilt's work involves applying cognitive and metacognitive boundaries to the user model, such as limiting the model to maintain at most seven pieces of information and forgetting the structure of previously visited pages.

5 Issues with previous research

Evaluating the usability of user interfaces is a constantly evolving and ever growing area in the field of computer science. Capturing usability data, analyzing and interpreting the data to identify usability issues, and critiquing the interface to suggest solutions or improvements to mitigate the problems is becoming integral to web systems development [26].

In the realm of computer science, predictions are made by analyzing user models that contain specific information about the behavior of users and types of users that frequent a web site. However, these user models do not take the users' goals or cognitive and metacognitive processes into consideration and base the results solely on statistically significant data. For example, clustering algorithms create distinct, typed groups of users or user sessions based on profiles, important episodes, traversal behavior, or external characteristics. These clustering algorithms ignore the users' goals, isolating potentially troublesome episodes, and various other attributes involved with creating and managing predictive cognitive user models. We believe that the users' goals and ability to retain and use information must be taken into consideration. As a start, we can analyze the sessionized access log and attempt to better understand the users' cognitive and metacognitive process, but we must also push to integrate the user models found within both fields [2].

Determining the effectiveness of a user interface or application includes analyzing multiple attributes and characteristics about the web site and web site's users. The notion of effectiveness in computer science has yet to include the cognitive and metacognitive processes of the user or the notion of user goals as seen in [2, 3, 5].

The majority of the research published in cognitive psychology with respect to applying predictive user models to graphical user interfaces consists of understanding the cognitive and metacognitive processes involved with various interactions. This research, as mentioned before, has led to the development of accurate predictive user models. However, graphical user interfaces have evolved far beyond basic menu interaction. While we can understand the dynamic of attention, perception, knowledge and learning with menus by human users through the use of cognitive user models, we do not fully understand the breadth or depth of cognitive processes involved with human users while performing complex tasks within a complex application.

There is a noticeable divide between computer science and cognitive psychology, the former lacking a thorough understanding of a users cognitive and metacognitive process and the latter lacking a holistic representation of graphical user interfaces and the significance of content and structure. However, both realms of science have managed to generate effective predictive user models which have been proven to generate accurate results. It should be stated that our assumptions and observations are not meant to understate the significance of the contributions made by previous research. Our observations have simply led us to believe that by integrating these user models we can create a more robust, more effective methodology.

6 Combining the User Models

We propose creating CBMGs of sessions created by users who have been clustered based on productions and goals found within a given task. These Production/Goal CBMGs, or PG-CBMGs, will show:

- The transition frequencies of the group's constituents as they traverse through the website
- The productions involved with reaching a given goal
- The average staying time per page found with a mapped production

The PG-CBMGs can be used as a model for future studies when there are modifications or additions to a user interface. For example, assume that modifications are made to an interface that consists of five pages. Given the PG-CBMGs for each user type we can determine how quickly, on average, each user type will complete certain defined goals.

To fully understand how this hybrid model will work, assume that the CBMG shown before in section 3.3 depicts these five pages and five states. Next, assume that a production, such as selecting an action provided by the website's user interface, has been mapped to the transition from page one to page four, and the goal is achieved by transitioning from page four to page five. Figure 2 allows us to visualize this example.

If we were to create a programmatic implementation to generate a series of PG-CBMGs that included all pages, transition frequencies, average staying times, productions, and goals for a given web application, we would have a predictive model based on user behavior derived from access logs. Creating this model would allow evaluators to answer questions such as:

- Are there too many (more than 4 $\pm$) steps (either actions or pages) involved with a given production
- Are the user's actions including the necessary productions to reach a goal
- Is the average staying time significantly higher for some productions
- Should some goals be merged, and others divided

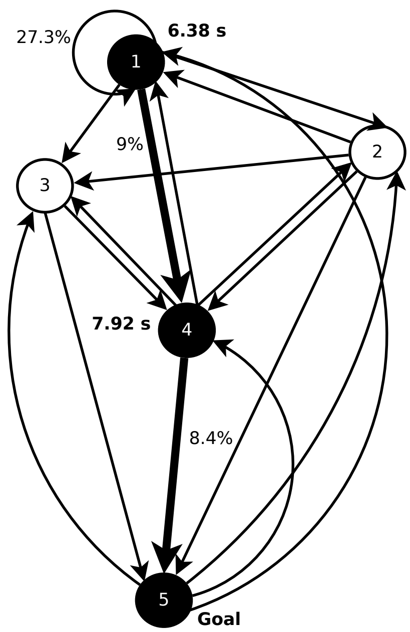

Figure 2: Production/Goal CBMG

Furthermore, we could break apart each node and annotate the individual latencies belonging to the perceptual, motor, and cognitive processes involved within each page. We could further extend the functionality of the methodology by creating a movie or interactive presentation showing the evolution of the PG-CBMG as the user traverses through various mapped productions to reach their goal. We believe this would not only create a better predictive cognitive user model that could be applied to any web application, but that it would also provide a tool that could help evaluators improve the effectiveness of the user interface by better understanding the cognitive and metacognitive processes of the users.

7 CryptoDB.org

For our case study, we wish to analyze how web users interact with the strategies workspace found within CrytpoDB.org. CryptoDB is an integrated genomic and functional genomic database for the parasite Cryptosporidium. CryptoDB integrates whole genome sequence and annotation along with experimental data and environmental isolate sequences provided by community researchers. The database includes supplemental bioinformatics analyses and a web interface for data-mining [27]. CryptoDB can be considered a fully functioning web application that interfaces massive datastores to assist web users in querying against and discovering information. The strategies workspace is a module within the web application that allows web users to manipulate, share, and save complex strategies. Strategies consist of steps where the web user is allowed to unionize, intersect, or subtract results from multiple queries in order to target specific sets of information to use within their research.

One of the issues the developers for CryptoDB have is determining whether or not the application is useful to the web users. That is, the developers do not have the means to determine whether using the strategies workspace allowed the web users to obtain their results, or reach their goals, more effectively, efficiently, and with few errors. While CTEGD, the organization charged with developing and maintaining CryptoDB, holds yearly workshops that allow them the chance to ask the web users questions directly, there is no clear way to determine the effectiveness of the strategies workspace with access logs alone.

Another issue the developers have is determining whether the novice users progress to intermediate and advanced users. They also face the issue of properly typing users based on the users' navigational patterns and behavior. In the past, researchers have provided sessionized access logs to the developers of CrytpoDB to have them manually type small sets of user sessions based on their behavior. With these annotated data sets, clustering algorithms were able to group the remaining sessions fairly accurately and derive results that allowed the developers to answer particular questions directly [1]. One of the questions the developers asked was whether a recent revision to the web application increased the transition frequencies of three different types of users, novice, intermediate, and advanced, of certain paths within the web application. The assumption being that if the transition frequency increased, more users were traversing the paths the developers wanted them to take, and thus the revision could be considered effective.

There are several issues with this approach, but the most notable is that effectiveness was determined based on what the developers thought the users should be doing. However, effectiveness should be defined based on the users' ability to reach their goals efficiently, and with as few errors as possible. It is also easy to see that the evolution of the user should also happen quickly and easily. With the notion of productions and goals as found in [2, 3, 23, 5, 4, 24, 6], we could easily redefine user types based on the productions found within user behavior as users attempt to achieve their goals, as well as the errors involved when attempting to reach the goals.

8 Proposal: Case Study

The proposal involves two separate studies. the first is to determine the effectiveness of tutorials and derive productions and errors involved with reaching a goal, and the second is to create a cognitive user model by typing the users then creating PG-CBMGs from the sessionized access log. The PG-CBMG can be used to make predictions of how well a user type would perform on a different system.

For the first part of the study, we will have two sets of users who are familiar with the domain, biology students preferably, to perform actions within the strategies workspace to meet a particular goal. This will be a think-aloud study where the web users will explain what they are thinking at every step while we capture video and audio of the procedure. The first set of users will perform a task to reach a desired goal, while the second set of users will watch a tutorial on how to use the strategies workspace before attempting to reach a desired goal. We will analyze the audio and video as well as the session information from each study to capture the staying time and path traversals of each user from each group. We will also annotate the sessions by assigning production names to the episodes with which they correspond, a process we will refer to as "production mapping". In other words, if we determine that production A consists of traversing from page X to page Y to page Z, we can map this episode to production A if found within a session. This is contingent upon the fact that each production will have a unique sub-path. If, however, separate productions share the same sub-path then we will have to rethink our approach for mapping productions to sub-paths.

We will then analyze the tutorial to determine which productions are shown to the viewer explicitly. Next, we will analyze the sessions created by the users to extract average staying time per page. Given this information we will then compare the sessions created by the first group to the sessions created by the second group to determine if watching the tutorial helped the second set of users reach the goal more effectively. The effectiveness of a tutorial could thus be measured by time, the percentage difference in total staying time, by error rate, the percentage difference in error rate, and degree of production existence, the percentage difference in the number of productions applied. The proposed equations for determining the effectiveness of a tutorial are shown below.

avg_{gr^a,go^x} = average time spent by all users in group a to reach a goal x

$error_{gr^a,go^x}$ = average error rate by all users in group a to reach goal x

$prod_{gr^a,go^x}$ = average number of productions by all users in group a to reach goal x

$E_{time} = avg_{gr^a,go^x}/avg_{gr^b,go^x}$ = the percentage increase or decrease of average time spent by all users within two groups to reach goal x.

$E_{error} = error_{gr^a,go^x}/error_{gr^b,go^x}$ = the percentage increase or decrease of average number of errors shown by all users within two groups while reaching goal x

$E_{prod} = prod_{gr^a,go^x}/prod_{gr^b,go^x}$ = the percentage increase or decrease of average number of productions shown by all users within two groups while reaching goal x.

$OE = 1.0 - (E_{time} + E_{error} + E_{prod}/3)$ = Overall Effectiveness of a tutorial. A positive overall effectiveness thus shows that the tutorial helped, a negative number shows that it hindered.

Next we would intersect two lists. The first list would consist of all observed productions from the first group, the second list would consist of all observed productions from the second group. This intersected list would contain productions known without watching the tutorial. We would then remove any productions from the list that did not directly assist the user in reaching their goal. We could then add these productions to the tutorial in an attempt to make the tutorial more effective.

The second part of our study is to type the users based on their behavior. For example, an advanced user may apply all the correct productions to reach the goal, whereas a novice user may apply several incorrect productions and did not reach the goal. This will have to be a supervised process where the site administrators determine what is novice behavior and what is advanced behavior, but the study will not be blindly guided as it was in our research found in [1] where the evaluators were asked to subjectively group the users based on guidelines created by the administrators alone. In supervising this process, we will generate distinct, typed groups of users. We will then merge the sessions of each group of users to form a PG-CBMG per group. The PG-CBMGs, as defined in the previous section, could be used to help the evaluators answer specific questions about the user interface before making significant changes. The PG-CBMGs could then be used to test the user interface once the changes are made.

9 Conclusion

In computer science, web characterization and user modeling has involved including various easily derived attributes. The user models can be created from parsing and sessionizing web server access logs, and generate viable information. As we have mentioned in our research, however, these models are limited to information annotated or created based on the evaluators' assumptions.

In cognitive psychology, cognitive user models are created based on questionnaires or observations made from analyzing the users' actions. Predictive cognitive models integrate a users cognitive and metacognitive process to generate accurate predictions of user behavior in a variety of scenarios.

Computer applications have become integral to our daily lives, and their complexity continues to grow. While the models created in the respective scientific fields have yielded positive results, a more robust, more effective model must be created. By merging models found within both fields, we feel that we can generate such a model. We proposed to test our ideas by combining ACT-R with CBMGs to create PG-CBMGs and applying the hybrid technique to a case study involving a live, intricate web application. We believe that by focusing our attention on productions and goals within a model created from access log data, we can create cognitive user model that can be applied to the majority of applications in existence.

There are several other models found within both fields of science, and thus a more effective merged model may be found elsewhere. One idea is to explore characterizing and clustering web users based on Deep Link Analysis metrics to create a stronger attachment between the web users and the information found within an access log before applying techniques found within cognitive user models [10]. Another idea is to apply a cognitive user model to a Time Series Analysis Selected Episode Graph (TSA-SEG). TSA-SEG is a methodology that first clusters user sessions based on their session level similarity then generates a visualization to show the changing transition frequencies of each cluster type over several time windows [1]. TSA-SEGs allow evaluators to determine the effectiveness of a website revision by observing the changing transition frequencies between modules within a website. For example, if an evaluation of a website yielded two distinct groups, and the groups were identified as being novice and advanced users, then the evaluators could focus their attention on important episodes found within the sessions and be provided with the transition frequencies throughout these episodes for each user type over several periods of time. Merging the TSA-SEG technique with ACT-R or GOMS may yield more viable results, allowing evaluators to not only observe the evolution of a user type, but understand exactly how a user evolved from one type to the next.

References

[1] K. Storm and E. Kraemer, Web site evolution: Usability evaluation using time-series analysis selected episode graphs, in *WSE '09: Proceedings of the Ninth IEEE International Symposium on Web Site Evolution*, pp. 27–38, 2009.

[2] R. W. P. Niels A. Taatgen, Addie Johnson and K.-P. L. Vu, *The Handbook of Human Factors in Web Design.* (, 2004), chap. 25.

[3] M. D. Byrne, International Journal of Human Computer Studies **55**, 41 (2001).

[4] H. P. Belkacem Chikhaoui, Evaluation of a contextual assistant interface using cognitive models, 2008.

[5] F. E. Ritter, D. V. Rooy, R. S. Amant, and K. Simpson, Providing user models direct access to interfaces: An exploratory study of a simple interface with implications for hri and hci, 2005.

[6] A. Das, A cognitive simulation model for novice text entry on cell phone keypads, 2007.

[7] S. Sengupta, Characterizing web workloads - a transaction-oriented view, in *IWDC*, pp. 102–111, 2003.

[8] M. syan Chen, S. Member, J. S. Park, and P. S. Yu, IEEE Transactions on Knowledge and Data Engineering **10**, 209 (1998).

[9] M. Eirinaki and M. Vazirgiannis, ACM Trans. Internet Technol. **3**, 1 (2003).

[10] D. Nicholas, P. Huntington, and A. Watkinson, Journal of Documentation **61** (2005).

[11] M. K. Zhenglu Yang, Yitong Wang, An effective system for mining web logs, pp. 40–52, Springer-Verlag Berlin Heidelberg, 2006.

[12] M. W. Heather A. Johnson, Medinfo .

[13] C. Kurz, H. Hlavacs, and G. Kotsis, Workload generation by modelling user behavior in an isp subnet, 2001.

[14] O. Nasraoui and H. Frigui, Extracting web user profiles using relational competitive fuzzy clustering, 2000.

[15] R. W. P. Niels A. Taatgen, Addie Johnson and K.-P. L. Vu, Improving the web site's effectiveness by considering each page's temporal information, pp. 47–54, Springer Berlin / Heidelberg, 2004.

[16] A. de Lucia, M. Risi, G. Tortora, and G. Scanniello, Clustering algorithms and latent semantic indexing to identify similar pages in web applications, in *WSE '07: Proceedings of the 2007 9th IEEE International Workshop on Web Site Evolution*, pp. 65–72, Washington, DC, USA, 2007, IEEE Computer Society.

[17] D. R. M. Scanniello, G. Distante, Using semantic clustering to enhance the navigation structure of web sites, in *WSE '08: Proceedings of the Sixth IEEE International Symposium on Web Site Evolution*, pp. 55–64, 2008.

[18] W. H. Hsu, J. Lancaster, M. S. R, and P. T. Weninger, Structural link analysis from user profiles and friends networks: A feature construction approach.

[19] B. Helfrich and J. A. Landay, Quip: Quantitative user interface profiling, 1999.

[20] D. A. Menascé, IEEE Internet Computing **7**, 89 (2003).

[21] A. De Lucia, G. Scanniello, and G. Tortora, Using a competitive clustering algorithm to comprehend web applications, in *WSE '06: Proceedings of the Eighth IEEE International Symposium on Web Site Evolution*, pp. 33–40, Washington, DC, USA, 2006, IEEE Computer Society.

[22] M. Rauterberg, Symbiosis of Human and Artifact (1995).

[23] M. Rauterberg, Amme: an automatic mental model evaluation to analyze user behaviour traced in a finite, discrete state space, 1993.

[24] R. S. Amant and T. E. Horton, Model-based evaluation of cell phone menu interaction, in *In Proceedings of the CHI04 Conference on Human Factors in Computer Systems*, pp. 343–350, ACM, 2004.

[25] C. T. Gene Lynch, Susan Palmiter, 5th Conference on Human Factors and the Web (1999).

[26] M. Y. Ivory and M. A. Hearst, ACM Comput. Surv. **33**, 470 (2001).

[27] CTEGD, Cryptodb: The cryptosporidium genome resource, `http://www.cryptodb.org`.