GOMS Theory and its use on the Instructional Design Process

Maricel Medina-Mora

George Mason University

Abstract

One of the most important phases during the instructional design process is the task analysis. During a traditional task analysis approach, a subject is observed in order to record the different tasks, actions and the time needed to perform a specific task. The results are used to define metrics and to optimize the workflow. With the introduction of computers as a mean to deliver instruction, the traditional approaches on task analysis needed to be redesigned in order to consider human computer factors. In 1983, a new model was developed in order to analyze the tasks carried out by a human being based on the cognitive information process: GOMS. In this paper, a brief description of the origin of this theory is presented, followed by the application of GOMS on the task analysis phase of the instructional design process. To illustrate the advantages of using GOMS in this phase, an example of a cognitive task analysis is presented using an automated tool named CogTool. Finally, an analysis of GOMS and its relation with the epistemological traditions of learning and the theories of learning is presented.

The origins of GOMS

GOMS stands for Goals, Operators, Methods and a set of Selection rules in order to accomplish a task. The original theory was developed by Card, Moran & Newell in 1983 who proposed a human processor model which involved perceptual, motor and cognitive user's actions. According to Abe and Ellington (2004), under that model, the brain can perform information processing operations such as comparing, matching and calculating. Because this model did not apply directly to task analysis or system design, Card et al. (Abe and Ellington, 2004) developed the engineering model, GOMS, which can be applied on a cognitive task analysis. As a result, the constraints on behavior, due to the task environment, can be mapped out and the user's knowledge about the task can be determined. Based on Kearsley (n.d), GOMS belongs to the theories of cognitive skills needed to accomplish a task using human computer interaction. It is based on a information processing framework, which proposes that in order to store information permanently in memory, the information must travel to three different memories: sensory memory, short-term memory and long term memory with separate perceptual, motor, and cognitive processing.

Based on Kieras (1997), a GOMS model includes the procedural knowledge a user must have in order to carry out a task on a system or device. GOMS has evolved into a family of models with techniques that predict usability. According to Kieras (1997), the simplest form of a GOMS model is the Keystroke-Level Model or KLM. The focus of KLM is to predict the time required to complete a task, also known as *time on task* metric. CPM-GOMS, developed by Gray, John and Atwood (as cited in Kieras, 1997), is another model which combines user's perception, cognition and motor process and map them into a schedule chart. The critical path is used to calculate the *time on task* metric. The last method described by Kieras (1997) is the

NGOMSL and stands for Natural GOMS Language. Through NGOMSL learning time and execution time are predicted. NGOMSL allows a software programmer to carry out a task by executing the actions described on the GOMS model. Those actions are the “how to do it” knowledge that a final user will need to do on the real system. Because GOMS contains design decisions from the user's point of view it is very valuable not only in the phases before the system design. It can be used as a basis for training and reference documentation because it already contains the descriptions of what users must learn.

To apply a GOMS technique, a task analysis of the system must be done in order to clearly identify what users can do on the tool. Once the goals are listed, a GOMS model can be generated. It is important to point that GOMS is not intended to be a substitute of the user interface analysis and design but provides enormous feedback to the designers about the amount, consistency, and efficiency of the procedures on the system (Kieras, 1997).

The task analysis constitutes the GOMS model and it should contain the goals, operators, methods, and selection rules. A goal describes “what the user wants to accomplish” on the system. A goal is also defined as “a symbolic structure that defines a state of affairs to be achieved” (Abe and Ellington, 2004, p. 3). Usually, subgoals must be already accomplished in order to reach a more general goal. According to Kieras (1997), a goal should be defined as an action-object pair: <*verb noun*> (e.g. WRITE-PAPER) and if the task includes complicated actions, they may be included on the task description. The operators describe the user actions (e.g, RIGHT-CLICK,). The difference with goals is that operators are executed but goals are accomplished. The methods are sequences of steps executed to accomplish a goal and contain an external or mental operator. Finally, selection rules are used to identify the appropriate method to accomplish a goal. If there are more than one method to accomplish a goal, then the goal should be decomposed into more specific goals, one for each method.

Furthermore, a GOMS model should include the task description and task instance. The ***task description*** describes a general task in terms of the goal. Basically, the task description includes the list of parameters, or preconditions, that the methods will require in order to perform the task. The ***task instance*** is a description of a specific task with specific values. This task instance can be described as scenarios of the task description.

GOMS in the instructional design process

The task analysis phase is one of the most relevant phases on the instructional design process. During this phase, the objective is to determine what the learner needs to know or accomplish after the instruction. As previously mentioned, GOMS already contains the description of what users should do in order to reach a goal. Therefore, using GOMS during this process will provide the inputs for training materials and furthermore, it can be used to analyze the learnability factors of the system and provide feedback to the system designers to consider a redesign on the user interface.

Jonassen, Tessmer, and Hannum (1999) analyzed GOMS and its use as a technique during the task analysis phase on the instructional design process. During a task analysis, an analyst documents users view of a task according to their natural goals, users decomposition of a task and the natural steps that are in the users methods (Kieras, 1997). When there is not a possibility to systematically gather behavioral data, the analyst can make assumptions of the process based on what the system allows the user to do. Current system's users provide valuable information regarding how they view a task; however, based on historical data about cognitive psychology (Kieras, 1997), very often users do not really know their own goals, strategies and general mental processes. Therefore, there is a difference between what users *think* they are doing and what they are *actually* doing, and what they *should* be doing. The best approach to obtain information from users is to ask them to perform a task and record what they are doing without interfering their work. Another consideration is that users can be performing a task without taking advantages of features that the system may already provide. This may be due to different reasons such deficiencies on the training process, low quality documentation or bad system design. The role of the analyst is to discover the causes. Is it the system being sub-optimal used or users know how to use it correctly? Furthermore, the most important goal is to determine the system's learnability. System's learnability is defined by Lew et al. (2007, p. 3) as the “ease with which one learns to operate a given system effectively”. The learnability factor is measured based on the time or effort that it takes to a user to get used to the system and its operation. Additionally, it includes how easy is to remember some operational steps in order to complete a task.

Because the analyses provide quantitative and qualitative information about the different tasks, they are both used on the instructional process. Training programs and help systems can be built based on the description of the knowledge needed to perform a task, producing a task-oriented document (Hochstein, 2002). Therefore, the user will know what he can do on the system (*goals*), the different actions to be performed (*operators*), different ways to achieve them (*methods*), and when to used them (*selection rules*).

The evaluation of the cognitive process during the execution of a task is another important factor during task analysis. Jonassen et al. as cited on (Lee, 2004) define cognitive task analysis as “model the cognitive process that a learner takes on when he/she performs certain task”. There are several software tools that help to describe a cognitive task analysis (Lee, 2004; Baumester, Bonnie & Byrne, 2000). In this paper, CogTool is presented as one example of these different tools. This tool was chosen because it is strongly based on GOMS techniques, it is totally free and it was available for downloading at the date of this paper.

CogTool

CogTool has been developed at the Human Computer Interaction Institute of Carnegie Mellon University (The CogTool Project, n.d). The tool provides an easy way to create KLM-GOMS models of skilled users performance behavior through the storyboard designs. In order to create the storyboard, the CogTool user must include the different screen shots on the tool. Each screen shot becomes a *frame*. On each different frame "hot-spots" or *widgets,* are created to simulate the user interaction. The widgets can be buttons, textboxes, listboxes, checkboxes, and any other interactive elements between the software and the user. Once this is done, the user links each frame to another frame through different kind of *transitions.* The final step is to model the task and record the steps needed to complete the task, this is done automatically by interacting directly with the fames. CogTool automatically places different "think" operators that serves to approximate cognitive processing time. CogTool generates the model on a KLM-like language which is executed through an ACT-R cognitive architecture, a theory for simulating and understanding human cognition (ACT-R, n.d.), and produces the performance prediction.

The example chosen to create a GOMS model through CogTool is the task "Write a reflection" using Blackboard, a Web-based course-management system. In one specific course, students have to write a reflection every week accessing Blackboard. During a cognitive task analysis the objective is to determine how long it will take to a student to complete this task. For an instructional designer, this exercise will help to evaluate the different steps, verify correctness of a instruction material and provide feedback to the product development team if the designer encounters usability issues that will prevent the effective learning and use of the tool. In this specific example, students must navigate through nine different screens to reach their goal (see Figure 1). The lines that link each frame on Figure 1, are the transitions mentioned above. The orange orange rectangles on each frame are the widgets. These widgets indicate that the user will need to interact with that specific element on the screen in order to move to the next frame or window. Figure 2 shows the script generated by CogTool that shows the different navigation steps. One relevant aspect of CogTool is that it automatically generates the cognitive action of thinking which, is part of the operators on the GOMS model. Each line on the script represents a step toward the final goal. Once the script is created, the latest step on CogTool is to compute the time on task prediction. Note that on Figure 3, the value calculated is 305.38 seconds. This value was generated according to the number of characters typed while entering the login, password and message textbox on Blackboard and also on specific times defined to read (think operator) and enter the response for the reflection. The estimated time on task will vary if the number of key strokes are changed. Finally, Figure 4 shows the ACT-R model generated by CogTool which contains the different operators need to complete the task. This operators include motor, visual and procedural actions. The graphic also shows how long it takes a user to complete a specific action. As can be seen, an instructional designer gets a lot of information through a cognitive task analysis by using an automated tool as CogTool. This information can be used to suggest reduction on the number of steps to complete a task, propose alternative user interfaces, and verify correctness of instructional materials such manuals or help files. Additionally, a deeper analysis can be carried out in order to evaluate the effects of visual saliences on the cognitive model of the user. Finally, it highly reduces the costs on usability testing because there is not need to use real users in order to do the cognitive task analysis and different prototypes can be generated and analyzed before deployment of final product.

GOMS, Epistemological Traditions and Learning Theories

Learning is the process in which an individual, machine, social groups, or animal acquires new knowledge and skill. Once the learning has occurred, a change in behavior and the way of doing different tasks also changes.

Because learning involves knowledge acquisition, it is important to recognize the different sources of knowledge and the epistemological traditions for each one. According to Driscoll (2005) there are three major epistemological traditions: pragmatism, objectivism and interpretivism. These three traditions serve as base for the theories of learning mentioned by Driscoll. On the pragmatism tradition, both experience and reason are the source of knowledge; but experience is the source on the objectivism tradition; and on the interpretivism tradition the knowledge comes from the use of reason. Based on these definitions, it can be said that when a person uses a system or device, the knowledge is obtained through a sensory experience. In this case, the knowledge falls into the objectivism tradition because the reality comes from outside the learner and its known through the continuous use of the tool.

On the other hand, different theories have been defined to identify how people learn. These theories have been grouped according to its epistemological tradition. A learning theory is defined as “a set of constructs linking observed changes in performance with what is thought to bring about those changes” (Driscoll, 2005, p.9). One construct is the memory which is the cornerstone on the cognitive perspective of learning and these theories are related to the objectivism tradition according to Driscoll. The cognitive information process, which views the learning as a transformation of information from one stage of memory to the other stage, plays a significant role on the cognitive perspective of learning. The memory stages are the sensory memory, working memory and long-term memory. GOMS's authors also viewed that the interaction between a computer and human being also used these three different stages on memory. When a person interact with a computer, he receives inputs from the different devices (e.g. monitor, speakers) which are stored on the sensory memory. The user defines the goals according to those inputs a generates a response, which is a set of operators to perform the goal. If the user already knows the system, he retrieves information from long term memory, in order to apply the correct methods according to the set of rules. This is how the GOMS model is created based on a cognitive information processing approach. However, because GOMS also focuses on user behavior, it has been also assigned as a behaviorist theory (Learning Theories, n.d.). Based on the definitions about learning theories an epistemology, I do not consider GOMS a learning theory by itself but a technique used to analyze the human computer interaction process. It uses not only discover the cognitive model of an user interacting with a system but it also provides the mechanism to estimate learning time, time on task and probability of errors. It also demonstrates that the information processing system plays an important role on the human computer interaction.

References

Abe, Crystal; Ellington, Beth (2004, August). Task analysis and human-computer interaction: approaches, techniques, and levels of analysis. *Proceedings of the Tenth Americas Conference on Information Systems.*

ACT-R (n.d.). Retrieved on November, 2007, from Carnegie Mellon University, Department of Psychology Web Site: http://act-r.psy.cmu.edu/

Baumester, Lynn K., Bonnie, John E., Byrne, Michael D.(2000) A comparison of tools for building GOMS models. *Proceedings of the SIGCHI conference on Human factors in computing systems*. (pp. 502-509) Netherlands: The Hague.

Driscoll (2005). *Psychology of learning for instruction* (3rd edition). Pearson

Hochsten, Lorin (2002). *GOMS*. Retrieved October 25, 2007, from University of Maryland, Department of Computer Science Web Site: http://www.cs.umd.edu/class/fall2002/cmsc838s/tichi/printer/goms.html

Jonassen, David H.; Tessmer, Martin; Hannum, Wallace H. (1999). *Task Analysis Methods for Instructional Design*. N.J: Lawrence Erlbaum Associates, Inc

Kearsley, Greg. *Explorations in Learning & Instruction: The Theory Into Practice Database* (n.d.). Retrieved September, 2007, from http://tip.psychology.org/

Kieras, David (1997). A Guide to GOMS Model Usability Evaluation using NGOMSL. In Helander, Martin G.; Landauer, Thomas K.; Prabhu, Prasad V. (2nd Ed.), *Handbook of human-computer interaction* (pp.733-766). Amsterdam: North Holland Elsevier.

Law, Effie Lai-Chong; Blazic, Borka Jerman; Pipan, Matic (2007, September). Analyses of user rationality and system learnability: Performing task variants in user tests. *Behaviour & Information Technology*, 26 (5) (pp. 421-436).

Learning-Theories (n.d.). Retrieved on October, 2007 from http://www.learning-theories.com

Lee, Y. (2004). Software review: Review of the Tools for the Cognitive Task Analysis. *Educational Technology & Society*, 7 (1), 130-139.

Vierzi, Robert A. (1997). Usability Inspection Methods. In Helander, Martin G.; Landauer, Thomas K.; Prabhu, Prasad V. (2nd Ed.), *Handb*http://www.cs.cmu.edu/~bej/cogtool/index.html*ook of human-computer interaction* (pp.705-732). Amsterdam: North Holland Elsevier.

Figure Caption

Figure 1. CogTool design for the task "Write a reflection" using Blackboard.

Figure 2. CogTool script generated for task "Write a reflection" using Blackboard.

Figure 3. CogTool generated time on task prediction for task "Write a reflection" using Blackboard.

Figure 4. CogTool ACT-R model generated for task "Write a reflection" using Blackboard.