Evaluation of a Contextual Assistant Interface Using Cognitive Models

Belkacem Chikhaoui, Hélène Pigot

***Abstract*—Cognitive models allow predicting some aspects of utility and usability of human machine interfaces, and also simulating the interaction with these interfaces. The action of predicting is based on a task analysis which analyses what a user is required to do in terms of actions and cognitive processes to achieve a task. Task analysis facilitates the understanding of the functionalities of the system to be modeled. Cognitive models are part of the analytical approaches that do not make necessarily appeal to the user during the interface development process. This paper presents a study about the evaluation of a human machine interaction (HMI) with an interface of a contextual assistant, using ACT-R and GOMS cognitive models. It shows how these techniques may be applied in HMI evaluation, design and research, emphasizing on the task analysis in one side, and on the time execution of tasks in the other side. In order to validate and support our results, an experimental study of user performance, during the interaction with the contextual assistant interface is conducted at the DOMUS laboratory. The results of our models show that both models GOMS and ACT-R give good to very good predictions of user performance at the task level as well as the object level, our results are very close to those obtained in the experimental study.

***Keywords*—HMI, interface evaluation, cognitive modeling, user modeling, user performance.

I. Introduction

THE evaluation of Human Machine Interfaces (HMI) is becoming increasingly important, while the development of interfaces presents some challenges; their evaluation needs rigorous methods to ensure they fulfill the initial specifications and the quality of accessibility and usability of these interfaces [1], [2]. Analytic evaluation techniques allow us to predict, among others, user performance, time execution of tasks, how a design will perform and to explain the performance of an existing interface [3]. Analytical approaches are not based directly on the user performance, but rather, on the automated examination of interfaces using well-defined structures and rigorous analysis techniques [4]. The most well-known methods of these techniques are cognitive models particularly ACT-R cognitive architecture [5] and GOMS family [6].

The human machine interaction should be resumed by the actions of pushing buttons displayed on a screen. According to this approach, the interaction process implies three human components, which must be taken in account. The first component is perceptual. In our case the human perceives the signal in a visual manner. The second one is cognitive. Here the human retrieves in his memory the object required and reasons to satisfy specific goals. The third one is motor and necessitates pressing on the selected button. The most important challenge of cognitive models is their capability to simulate the three components: perceptual, cognitive and motor during the task execution process.

In this study, we aim to evaluate the interaction with an interface of a contextual assistant developed for cognitively impaired people. This application assists people while preparing meals in their kitchen by using cognitive assistance [7]. Due to the related population and the kind of errors they commit, we need to take in account the cognitive part involved in the HMI. We then use a powerful analytical methods based on cognitive models, emphasizing the cognitive analysis of the tasks in one side, and the time execution of tasks in the other side. Our analytical evaluation is based on two methods, the first one is the cognitive architecture ACT-R [5] in which the interaction is decomposed in rules simulating the cognitive behavior of a human using the contextual assistant, and the second one is the GOMS (Goals, Operators, Methods and Selection rules) model which is a formalized representation that can be used to predict task performance [8]. The GOMS model is a way in which users achieve goals by solving subgoals in a divide-and-conquer fashion [9].

In order to create an effective evaluation, an empirical study is conducted at the DOMUS laboratory with 10 normal persons. The results of our models are compared with those obtained in the experimental study.

After introducing an overview of the cognitive architecture ACT-R, the GOMS model and the contextual assistant, we present the task to be analyzed and modeled and the experiment study. The models developed are then introduced and the results of the simulation are compared to the results obtained in the experimental study.

II. Background

In this section we present an overview of the cognitive architecture ACT-R, the GOMS model and then we introduce the contextual assistant application and the interface with which users interact.

A. Cognitive Architecture ACT-R

The cognitive architecture ACT-R is built to simulate and understand human cognition [5], [10]. It consists of a set of modules integrated through a central production system. ACT-R is an hybrid architecture that combines two subsystems: symbolic system including semantic and procedural

Belkacem Chikhaoui is with the Domus Laboratory, Computer Science Department, Faculty of Science, Université de Sherbrooke, Sherbrooke QC J1K 2R1 Canada, email: belkacem.chikhaoui@usherbrooke.ca

Hélène Pigot is with the Domus Laboratory, Computer Science Department, Faculty of Science, Université de Sherbrooke, Sherbrooke QC J1K 2R1 Canada, email: helene.pigot@usherbrooke.ca

knowledge, and subsymbolic system evaluating knowledge activations. The subsymbolic system assigns activations to chunks (semantic knowledge) and rules (procedural knowledge), which helps to choose the more predominant knowledge available at a specific time. In ACT-R the perceptual and motor modules are used to simulate interfaces between the cognitive modules and the real world. The perceptual modules allow the model to attend to visual and aural stimuli, while the motor modules are responsible for preparing and executing basic motor actions, such as key presses and mouse movements [11], [12].

The visual module that is part of the perceptual modules, is decomposed on two subsystems, the positional system (where) and the identification system (what), that work together in order to send the specified chunk to the visual buffer. The positional system is used to find objects. When a new object is detected, the chunk representing the location of that object is placed in the visual-location buffer according to some constraints provided by the production rule. The identification system is used to attend to locations which have been found by the positional system. The chunk representing a visual location, will request the identification system to shift visual attention to that location. The result of an attention operation is a chunk, which will be placed in the visual buffer [11], [12].

The motor module contains only one buffer through which it accepts requests. Two actions are available in ACT-R, to click with the mouse or press a key on a virtual keyboard.

B. GOMS model

GOMS [6], [8], is an acronym for Goals, Operators, Methods and Selection rules. It is a formalized method used to predict task performance. A GOMS description consists of these 4 elements:

1) Goals: The user's goals describing what the user wants to achieve.

2) Operators: The basic actions that the user must perform in a lowest level of analysis, in order to use the system [9].

3) Methods: Methods are sequences of steps consisting of operators and subgoal invocations, that the user performs in order to accomplish a goal.

4) Selection rules: Selection rules choose the appropriate method, depending on the context when choice of methods arises [9], [8].

C. Contextual Assistant

The Contextual assistant is a framework developed to assist persons with cognitive disabilities [13], [14]. The aim is to foster autonomy in the daily living tasks, and particularly during complex cooking tasks, such as preparing spaghetti [15]. The cooking task is decomposed of steps displayed on a touch screen. The two first steps consist of gathering the utensils and ingredients necessary to the recipe (Figure 1). The other steps explicit the recipe using photo and video on the screen as well as information dispatched all around the kitchen. The contextual assistant is specifically designed to help people remembering the places where the objects are stored. To do so, the contextual assistant contains an interface called the object locator displaying the objects to search. When an object is pushed in the main interface, the contextual assistant looks for the location of that object in the environment, using techniques of pervasive computing, and indicates the location by highlighting the appropriate locker containing that object. In this study we simulate the first two steps of the spaghetti recipe. They consist of first, knowing the list of objects to gather, either utensils or ingredients, and then to use the object locator in order to find each object.

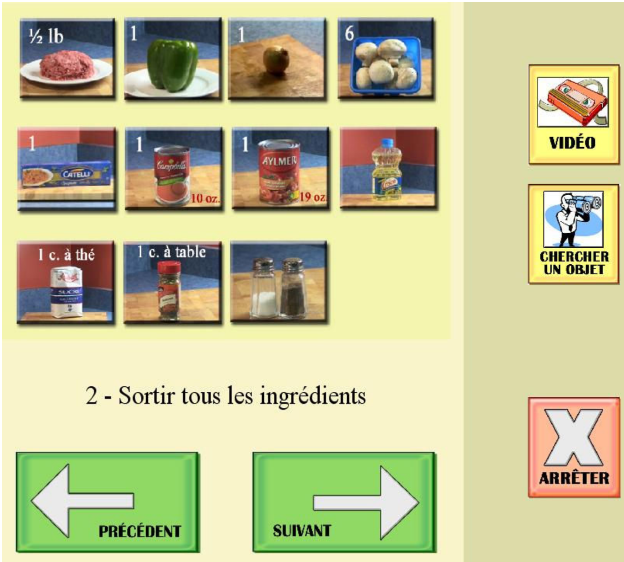


Fig. 1. Interface of the gathering ingredients task of the contextual assistant

The contextual assistant interface is displayed on a 1725L 17" LCD Touchscreen, with 13.3" (338 mm) horizontal and 10.6" (270 mm) vertical useful screen area. It is configured to 1024 x 768 optimal native resolution running Macintosh. The screen is fixed under a closet nearby the oven in order to be easily accessible and also protected against the cooking splashes.

III. Experimental Study

In this section we describe the procedure of the performance study conducted at the DOMUS laboratory, in terms of users, apparatus and applications used to perform this study.

A. Apparatus and Application

In order to measure the specific time of the interaction with the interface of the contextual assistant, we measure the time that users take to decide which object they want to get out, and the time to push on that object in the interface of the contextual assistant.

The first action corresponding to "decide which object users want to get out", can be presented experimentally using a PDA (Personal Digital Assistant). To do so, we developed an application that displays to users, the name of the object to get out, in order to highlight the recognizing object phase involved in the cognitive processes, and this is applied for each object needed in the two tasks, gathering utensils and gathering ingredients.

The specified application is installed on a PDA and executed in parallel with the contextual assistant. The time taken to push each object on the PDA is recorded in a log file, this latter is recovered at the end of the experimentation.

B. Subjects

Ten students of the Sherbrooke university participate in the study. All subjects are male and their ages range from 27 to 32 years. The subjects have corrected vision with no other physical impairments being reported. All subjects have a good knowledge of computer science.

C. Method

The PDA is deposed at a distance of 15 cm from the touchscreen, participants remain standing at a distance of approximately 30 cm from the touchscreen during the entire test. Participants started with a practice stage in order to familiarize themselves with the interface.

When the test is started, participants start to look on the PDA to know the name of the object to be get out, and push the corresponding button of this object in the contextual assistant interface, by touching it with their index finger. Once the button is pushed, participants push the button representing the name of the object on the PDA and pass to the next object displayed on the PDA, and the experimentation will continue until the last object of the gathering ingredients task is reached. During the entire test, objects displayed on the PDA were presented to participants in a randomized order. This emphasizes the recognizing objects phase in the cognitive processes. Each participant accomplishes 5 trials, and each trial in the study requires achieving the two tasks gathering utensils and gathering ingredients. Each trial needs 25 actions "pushing button on the PDA" and 25 actions "pushing button on the contextual assistant interface". Altogether 2500 (10 participants x 5 trials x (25 actions x 2 interfaces (PDA and contextual assistant)) = 2500) actions are observed during the experimentation.

In our study, the action of getting out the objects from their locations in the environment is not modeled.

Table I shows the mean duration with the standard deviation for each object in the two tasks, over all participants in our study.

TABLE I
USER PERFORMANCE DATA ACROSS OBJECTS WITH MEAN AND STANDARD DEVIATION

Objects	Duration (s)	Standard Deviation (S)
LOOK-FOR-OBJECT (Utensils)	5.299	1,052
CAN-OPENER	2.291	0,717
COLANDER	2.966	0,786
MEASURING-SPOON	2.167	0,605
LADLE	2.847	0,829
SMALL-SAUCEPAN	1.980	0,371
WOODEN-SPOON	2.590	0,536
KNIFE	2.328	0,430
BIG-SAUCEPAN	1.779	0,308
CUTTING-BOARD	2.000	0,309
HELP-ME-TO-DO-THE-TASK (Utensils)	2.039	0,386
NEXT	2.142	0,540
LOOK-FOR-OBJECT (Ingredients)	1.955	0,265
PEPPER	2.448	0,825
SPAGHETTI	1.939	0,552
TOMATOES-BOX	1.794	0,377
GROUND-BEEF	2.491	0,591
ONION	2.021	0,484
TOMATO-SOUP	1.970	0,422
SALT-AND-PEPPER	2.490	0,481
OIL	1.965	0,348
MUSHROOMS	1.809	0,369
SUGAR	1.774	0,341
ITALIAN-SPICE	1.736	0,436
HELP-ME-TO-DO-THE-TASK (Ingredients)	2.432	0,614

IV. MODELING THE INTERACTION WITH THE CONTEXTUAL ASSISTANT USING ACT-R

In this section, we present the modeling process of the tasks involved in our study, which are gathering utensils and gathering ingredients, emphasizing on the perceptual and cognitive parts, using the perceptual motor modules of ACT-R.

A. Task analysis: gathering utensils and ingredients

We model the first two steps of the recipe, gathering utensils and gathering ingredients. The interactions with the touch screen are simulated without taking in account the time taken by the subject to pick up the objects in the environment. The two first steps require three subtasks (Figure 2). The first subtask consists of activating the object locator in order to locate each object required by the recipe. This is done by pushing the button "LOOK-FOR-OBJECT", which is displayed on the main interface of the contextual assistant (Figure 1). The second subtask is to locate each object, either utensils or ingredients, needed in the current step by pushing the button corresponding to the object in the object locator. The third task consists of coming back to the main interface of the contextual assistant in order to know the next step of the recipe. The tree decomposition is presented in Figure 2, to compare the tasks tree from the interface of Figure 1. The nodes in capital indicate the action to click on the named button, while the other nodes represent tasks to be decomposed.

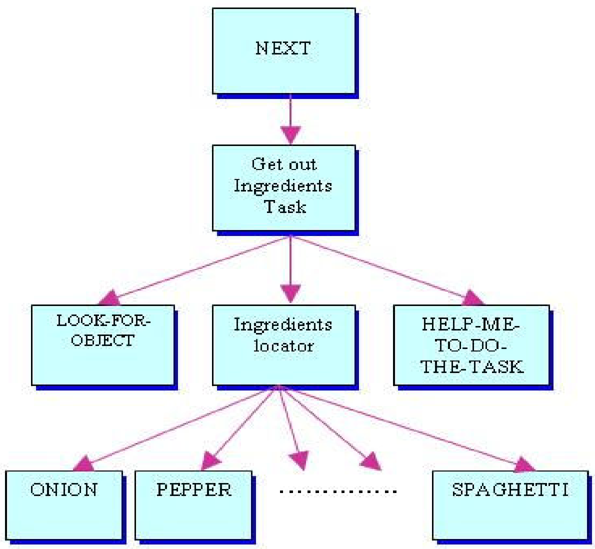


Fig. 2. Tree representing the gathering ingredients task

B. Gathering utensils and ingredients model

The model developed aims to simulate the IHM using the contextual assistant. The model is restricted to the two first steps of the recipe and only to the interactions with the touch screen. In that task, three different interfaces are involved, the interface of the object locator and the two of the contextual application, displaying the utensils and the ingredients needed in the recipe. The model uses ACT-R to emphasize the cognitive processes involved, when looking for an object and choosing the button to push. It is subdivided on three phases, the visual phase, the recognizing phase and the motor phase.

The visual phase consists of localizing the object to perceive and then identifying it. We consider that all buttons displayed on the screen are objects, either the buttons used to locate a utensil or ingredient, or the buttons to navigate in the interface. The first one is the button "LOOK-FOR-OBJECT" as described in Figure 2. Then, all the utensils (or ingredients) needed in the recipe are presented in the visual interface of ACT-R. Finally, to complete the current step of the recipe, the button "HELP-ME-TO-DO-THE-TASK" is presented, in order to come back to the main interface of the contextual assistant, and pursue the next step of the recipe. Each object of the interface is displayed at defined coordinates (x, y) on the screen. These coordinates specify the request made to the visual-location buffer of ACT-R, which creates a chunk representing the location of the specified object. After that, the identification system identifies the name of that object and creates a chunk which will be placed in the visual buffer. The location and identification phases last 185 ms [11], [12]. The objects are presented to the visual module of ACT-R by the mean of a list of all the objects (buttons of the interface) to be pushed on.

The recognizing phase begins when the chunk of the object is placed in the visual buffer. This phase implies to recover that specific chunk from the declarative memory. The result of this phase is a chunk that represents the object with some characteristics as color, localization on the screen, name, and kind of object.

The motor phase consists of activating the motor actions via a request to the motor buffer, in order to click on the object. The three phases are applied for each object displayed in the interface for the two steps of the recipe. The gathering utensils and ingredients model finishes, when the last object of the gathering ingredient task is reached.

In our ACT-R model, the interface of the contextual assistant is simulated using a virtual display based on a vertical list in the *Lisp* environment. The virtual display maintains a representation of each object used in the interface at a given time, by displaying its name surrounded by a red circle, which reflects the shift attention to that object, as shown in Figure 3.

The ACT-R model is developed using the ACT-R 6 environment. No noise is introduced in our model. No retrieval error is modeled in the recognizing phase. These restrictions lead to a deterministic model. Figure 4 shows in a very low detailed form, the execution traces of the ACT-R model of the visual encoding, shift attention, recognizing phase and motor actions. The visual-location request takes place at time 0.050 seconds, and the request to move-attention is made at time 0.100 seconds. The encoding needs still 0.085 seconds to be completed and store the chunk into the visual buffer. After this phase, a retrieval request is made on the retrieval buffer, in order to recover the specified chunk from the declarative memory, this phase will finish at time 0.397 seconds, and finally a request on the motor buffer starts at time 0.447 seconds.

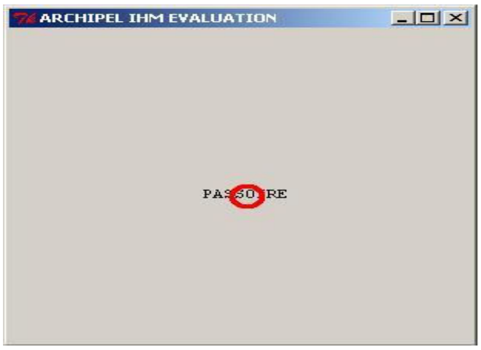


Fig. 3. Shift attention representation in the ACT-R model

```
0.000   GOAL        SET-BUFFER-CHUNK GOAL FIRST-GOAL REQUESTED NIL
0.050   PROCEDURAL  PRODUCTION-FIRED START-APPLICATION
THE SUBJECT STARTS TO LOOK FOR NEW OBJECT
0.050   VISION      SET-BUFFER-CHUNK VISUAL-LOCATION LOC1
0.050   GOAL        SET-BUFFER-CHUNK GOAL GET-OBJECT0
0.100   PROCEDURAL  PRODUCTION-FIRED ATTEND-USTENSIL
SHIFT ATTENTION TO A SPECIFIED LOCATION ON THE SCREEN
0.185   VISION      SET-BUFFER-CHUNK VISUAL TEXT1
0.235   PROCEDURAL  PRODUCTION-FIRED ENCODE-USTENSIL
ENCODING THE OBJECT AFTER VISUAL FOCUS
0.235   IMAGINAL    SET-BUFFER-CHUNK IMAGINAL OBJECT1
0.285   PROCEDURAL  PRODUCTION-FIRED FOUND-OBJECT
0.397   DECLARATIVE SET-BUFFER-CHUNK RETRIEVAL OBJECT1-0
0.447   PROCEDURAL  PRODUCTION-FIRED MOTOR-ACTION
```

Fig. 4. Example of execution trace of the ACT-R model

V. Modeling The Interaction With The Contextual Assistant Using GOMS

After analyzing the task to be performed as described above, the GOMS model can be started. The first two steps of the recipe, gathering utensils and gathering ingredients can be interpreted in the GOMS language by a principal method which can be defined as shown in Figure 5

For each task in our study, a method is defined following the concepts of GOMS methods in the definition of goals and subgoals. The GOMS model is based on a hierarchical representation of goals, in fact, the user achieves goals by solving subgoals [9], until reaching the basic operations called "operators" which can not be subdivided. The methods have a

```
Method_for_goal: Archipel Evaluation
        Step 1. Accomplish_goal: Select Utensils.
        Step 2. Accomplish_goal: Select Ingredients.
        Step 3. Return_with_goal_accomplished.
```

Fig. 5. Main method of the GOMS model

hierarchical structure, and a method may call for subgoals to be accomplished [16]. Figure 6 shows the hierarchy of subgoal invocations for the "Select Ingredients" method.

```
Method_for_goal: Select Ingredients
    Step 1.  Accomplish_goal: Select NEXT.
    Step 2.  Accomplish_goal: Select LOOK-FOR-OBJECT.
    Step 3.  Accomplish_goal: Select ONION.
    Step 4.  Accomplish_goal: Select PEPPER.
    Step 5.  Accomplish_goal: Select GROUND-BEEF.
    Step 6.  Accomplish_goal: Select MUSHROOMS.
    Step 7.  Accomplish_goal: Select TOMATO-SOUP.
    Step 8.  Accomplish_goal: Select ITALIAN-SPICE.
    Step 9.  Accomplish_goal: Select OIL.
    Step 10. Accomplish_goal: Select SUGAR.
    Step 11. Accomplish_goal: Select SALT-AND-PEPPER.
    Step 12. Accomplish_goal: Select TOMATOES-BOX.
    Step 13. Accomplish_goal: Select SPAGHETTI.
    Step 14. Accomplish_goal: Select HELP-ME-TO-DO-THE-TASK.
    Step 15. Return_with_goal_accomplished.
```

Fig. 6. Method for Select Ingredients task

The main method presented in Figure 5 constitutes the root of tree hierarchy, and all the other methods are generated automatically using the divide-and-conquer technique [9]. In our GOMS model, each object is defined as visual object and the select methods have the same form for all objects. Figure 7 shows the tree decomposition corresponding to the main method of the GOMS model.

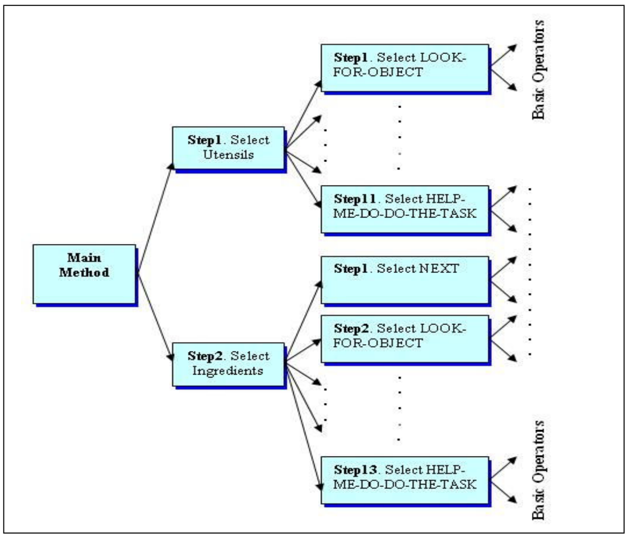


Fig. 7. Tree decomposition of the main method in the GOMS model

The duration of a step in the GOMS model can be defined as the sum of the production cycle duration which equals to 50 ms and the duration of all actions included in the body of the step [17], [18] such as key presses which take 280 ms to be performed.

Our model is executed using the GLEAN3 modeling tool [17].

VI. Comparison Of Results

We describe the performance of our models at two levels, the accuracy with which our models predict the overall duration of tasks, and the accuracy to predict the duration to push each object displayed in the interface.

A. *Object level performance*

Table II shows summary of user performance by object level and the ACT-R and GOMS model predictions. Values in parentheses represent the smallest and greatest value of user data observed for each object. Figure 8 shows the same data in a detailed graphical form. The two models GOMS and ACT-R give good approximations of user performance.

TABLE II
Comparison of user data, ACT-R and GOMS model predictions by object

Objects	User Performance (s)	ACT-R (S)	GOMS (S)
LOOK-FOR-OBJECT (Utensils)	5.299 (3.838 7.052)	2.230	4.250
CAN-OPENER	2.291 (1.308 3.676)	2.165	2.550
COLANDER	2.966 (1.703 4.130)	2.250	2.550
MEASURING-SPOON	2.167 (1.206 3.262)	2.250	2.550
LADLE	2.847 (1.434 4.143)	2.165	2.550
SMALL-SAUCEPAN	1.980 (1.351 2.652)	2.080	2.550
WOODEN-SPOON	2.590 (1.561 3.622)	2.250	2.550
KNIFE	2.328 (1.676 2.947)	2.165	2.550
BIG-SAUCEPAN	1.779 (1.284 2.330)	2.165	2.550
CUTTING-BOARD	2.000 (1.600 2.451)	2.165	2.550
HELP-ME-TO-DO-THE-TASK (Utensils)	2.039 (1.349 2.680)	2.165	2.550
NEXT	2.142 (1.289 3.082)	2.165	2.550
LOOK-FOR-OBJECT (Ingredients)	1.955 (1.384 2.362)	2.165	2.650
PEPPER	2.448 (1.159 3.847)	2.250	2.550
SPAGHETTI	1.939 (1.311 3.180)	2.165	2.550
TOMATOES-BOX	1.794 (1.265 2.382)	2.165	2.550
GROUND-BEEF	2.491 (1.634 3.643)	2.165	2.550
ONION	2.021 (1.391 2.906)	2.165	2.550
TOMATO-SOUP	1.970 (1.373 2.714)	2.165	2.550
SALT-AND-PEPPER	2.490 (1.546 3.227)	2.165	2.550
OIL	1.965 (1.389 2.544)	2.165	2.550
MUSHROOMS	1.809 (1.232 2.477)	2.250	2.550
SUGAR	1.774 (1.213 2.300)	2.250	2.550
ITALIAN-SPICE	1.736 (1.217 2.772)	2.250	2.550
HELP-ME-TO-DO-THE-TASK (Ingredients)	2.432 (1.265 3.654)	2.165	2.550

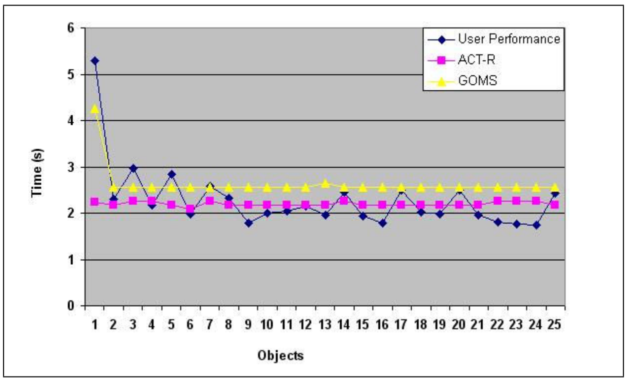


Fig. 8. User data, ACT-R and GOMS model predictions by object

According to Figure 8, the results of both ACT-R and GOMS models are very close, and have approximately the same predicted time values for several objects. Figure 9 shows the progression in accomplishing tasks for both ACT-R and GOMS models depending on the time progression. Since the procedure of prediction is applied for each object in the interface, the two models follow a linear model. This is supported by some scientific literature [18].

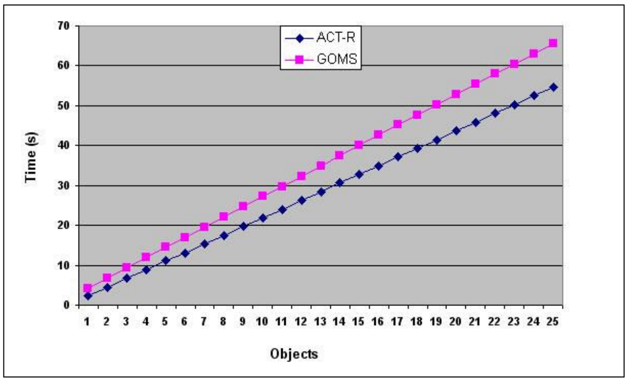


Fig. 9. Progression in accomplishing tasks for both ACT-R and GOMS models

B. Task level performance

Table III shows the user performance data and the ACT-R and GOMS model predictions in both tasks: gathering utensils and gathering ingredients. Figure 10 shows the same data in a detailed graphical form.

TABLE III
COMPARISON OF USER DATA, ACT-R AND GOMS MODEL PREDICTIONS BY TASK

Task	User Performance (s)	ACT-R (s)	GOMS (s)
Gathering utensils	28.290	24.050	29.750
Gathering ingredients	28.973	30.650	35.800

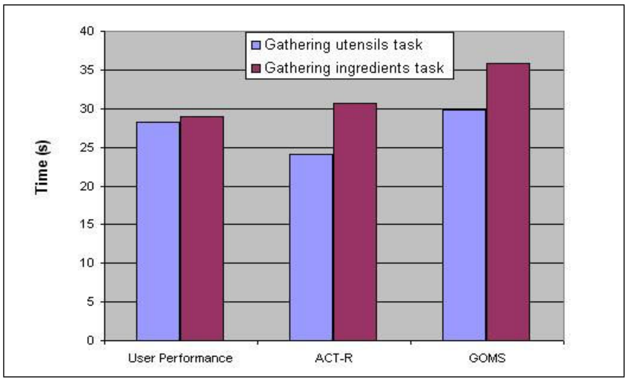


Fig. 10. User data, ACT-R and GOMS model predictions by task

The predicted time in both ACT-R and GOMS models is very close depending on the time progression of tasks of user performance, as shown in Figure 11.

VII. GENERAL DISCUSSION

The ACT-R and GOMS models we developed have proved robust and efficient. In fact, the results of both models are very close to the user performance data obtained in the experimental study. The GOMS and ACT-R models give good to very good predictions of time execution of tasks as well as objects as shown in Table II and Table III.

As shown in Table II, the object "LOOK-FOR-OBJECT (Utensils)" needs more time to be pushed using the GOMS model (4.250 seconds) than the ACT-R model (2.230 seconds), this significant difference can be interpreted by the fact that, the GOMS model includes mental operator at the beginning of the task gathering utensils. This mental operator takes 1.2 seconds to be accomplished. The same difference is observed in the predicted time of the object "LOOK-FOR-OBJECT (Ingredients)" with the GOMS model, this object necessitates more time to be pushed (2.650 seconds), which can be interpreted by the addition of a mental operator at the beginning of the task gathering ingredients.

Some differences in the predicted time of some objects using the ACT-R model are observed as shown in Table II, this is due to several rules such as visual processing when a new object is detected in the visual field, information retrieval and motor actions. The visual part in the ACT-R model is explicitly defined using requests on the visual buffers, unlike the GOMS model in which the visual part is implicitly defined. Both ACT-R and GOMS models do not take into account the location of objects displayed in the interface of the contextual assistant, unlike the experimental study in which the location of objects is included in the user performance data.

Our results show that, the evaluation of HMI designed for persons with cognitive disabilities, at a detailed low level is possible using cognitive modeling techniques, particularly ACT-R and GOMS models. During the development of the two models ACT-R and GOMS, we observed that the GOMS model gives more flexibility in modeling than the ACT-R model, which constitutes the intricate part in our study. Table III shows that both models ACT-R and GOMS give good approximations of user performance at the task level. The results of our models are considered suitable and correct, comparing them to the user performance data obtained in the experimental study.

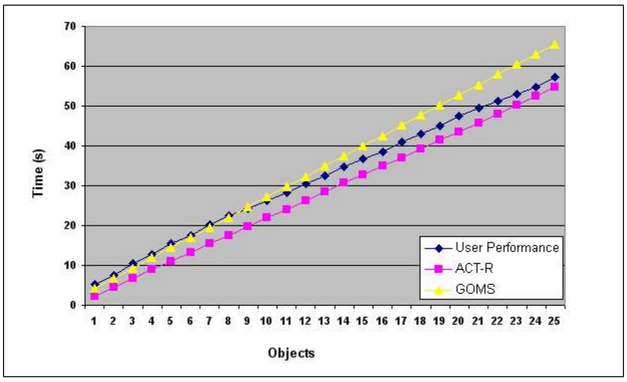


Fig. 11. Progress of time depending on progress in accomplishing tasks

VIII. CONCLUSION

This study empirically demonstrated that cognitive models are a powerful tool for evaluating interfaces and predicting user performance. The main goal of our study is to evaluate the interface of a contextual assistant by simulating the human machine interaction, focusing on the time execution of tasks. We used two efficient and powerful cognitive models to evaluate the specified interface, the first model is based on the cognitive architecture ACT-R and the second one is based on the GOMS model. The results show that the GOMS model can predict user performance at good level, and the ACT-R model can predict user performance at more detailed level and performs almost as well, which make our models powerful and realistic.

According to these results, the two models could be used to improve the design of the interface of the contextual assistant and to optimize it. Our study makes two main contributions, the first one is the evaluation of human machine interfaces designed for cognitively impaired people which constitutes a new study in this field, and the second one, is the use of cognitive models to evaluate these interfaces emphasizing on the cognitive processes involved during the human machine interaction. This first approximation of the cognitive process involved during human machine interaction, where only the cognitive action is resumed to the retrieval one helps us to validate the model, in terms of time required to interact with the contextual assistant interface.

IX. Future Improvements

Some improvements should be brought to our model. First, our model is deterministic and does not make errors. It should be extended to allow errors in the pointing actions such as: pushing an object several times before or after to look for the location of that object in the environment, or pushing an object instead of another one. These errors are essentially related to memory problems that may occur in the task modeling [19], [20] and during the interaction with the interface of the contextual assistant. Second, since the contextual assistant is designed to assist cognitively impaired people in smart homes, it would be interesting to do some experimentations with this population, which allows us to study the behavior of our models in real situations and to evaluate their performance and effectiveness. Finally, the action of searching an object is resumed to the HMI with the touch screen. The contextual assistant offers an interaction with the environment to help people recovering utensils and ingredients dispatched in the kitchen. It would be interesting in the future to model this part and simulate the movement of users picking up the objects in the kitchen.

References

[1] J. Nielsen and V. L. Phillips, "Estimating the relative usability of two interfaces: heuristic, formal, and empirical methods compared," *Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*, pp. 214–221, 1993.

[2] B. D. Eugenio, S. Haller, and M. Glass, "Development and evaluation of nl interfaces in a small shop," *AAAI Spring Symposium on Natural Language Generation in Spoken and Written Dialogue*, pp. 1–8, 2003.

[3] L. M. Leventhal and J. A. Barnes, *Usability Assessment*, pearson prentice hall ed., ser. Usability Engineering: Process, Products, and Examples, 2007, ch. 11, pp. 206–220.

[4] B. Yen, P. Hu, and M. Wang, "Towards effective web site designs: A framework for modeling, design evaluation and enhancement," *Proceedings of IEEE International Conference on e-Technology, e-Commerce, and e-Service (EEE05)*, pp. 1–6, 2005.

[5] J. R. Anderson, D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere, and Y. Qin, "An integrated theory of the mind," *Psychological Review*, vol. 111, 136-1060, 2004.

[6] B. E. John and D. E. Kieras, "Using goms for user interface design and evaluation: Which technique?" *ACM Transactions on Computer-Human Interaction*, vol. 3, no. ACM, pp. 287–319, 1996.

[7] H. Pigot, J. P. Savary, J. L. Metzger, A. Rochon, and M. Beaulieu, "Advanced technology guidelines to fulfill the needs of the cognitively impaired population," in *3rd International Conference On Smart Homes and health Telematic (ICOST)*, ser. Assistive Technology Research Series, S. Giroux and H. Pigot, Eds. IOS Press, 2005, pp. 25–32.

[8] D. E. Kieras, "Goms models for task analysis," *Handbook of task analysis for human-computer interaction, Lawrence Erlbaum Associates*, no. Diaper D, Stanton N, pp. 83–116, 2003.

[9] A. Dix, J. Finlay, G. D. Abowd, and R. Beale, *Human-Computer Interaction*. England: Pearson, Prentice-Hall, Inc, 2004.

[10] J. R. Anderson, N. A. Taatgen, and M. D. Byrne, "Learning to achieve perfect time sharing: Architectural implications of hazeltine, teague ivry (2002)," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 31, No. 4, pp. 749–761, 2005.

[11] M. D. Byrne, "Act-r/pm and menu selection: Applying a cognitive architecture to hci," *International Journal of Human-Computer Studies*, vol. 55, pp. 41–84, 2001.

[12] D. Bothell, "Act-r 6.0 reference manual," Working Draft, 2004.

[13] H. Pigot, J. Bauchet, and S. Giroux, *Assistive devices for people with cognitive impairments*, ser. The Engineering Handbook on Smart Technology for Aging, Disability and Independence. John Wiley and Sons, 2007, ch. 12.

[14] D. Lussier-Desrochers, Y. Lachapelle, H. Pigot, and J. Bauchet, "Apartments for people with intellectual disability: Promoting innovative community living services," in *2nd International Conference on Intellectual Disabilities/Mental Retardation.*, 2007.

[15] H. Pigot, D. Lussier-Desrochers, J. Bauchet, Y. Lachapelle, and S. Giroux, "A smart home to assist recipes completion." in *Festival of International Conferences on Caregiving, Disability, Aging and Technology (FICCDAT), 2nd International Conference on Technology and Aging (ICTA)*, 2007.

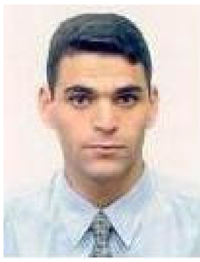
[16] B. E. John and D. E. Kieras, "The goms family of user interface analysis techniques: Comparison and contrast." *ACM Transactions on Computer-Human Interaction*, vol. 3, no. ACM, pp. 320–351, 1996.

[17] D. E. Kieras, S. D. Wood, K. Abotel, and A. Hornof, "Glean: A computer-based tool for rapid goms model usability evaluation of user interface designs," *Proceedings of the 8th ACM Symposium on User Interface Software and Technology*, no. ACM, pp. 91–100, 1996.

[18] R. S. Amant, T. E. Horton, and F. E. Ritter, "Model-based evaluation of expert cell phone menu interaction," *ACM Transactions on Computer-Human Interaction*, vol. 14(1), pp. 1–24, 2007.

[19] A. Serna, H. Pigot, and V. Rialle, "Modeling the performances of persons suffering alzheimers disease on an activity of the daily living," in *18th Congress of the International Association of Gerontology*, 2005.

[20] A. Dion and H. Pigot, "Modeling cognitive errors in the realization of an activity of the everyday life," in *Cognitio 2007*, 2007.



Belkacem Chikhaoui received the Eng. degree in computer science from the University of Boumerdès. Algeria, in 2004. He is currently pursuing the Master degree in computer science at the Université de Sherbrooke. Sherbrooke, QC, Canada. His research interests include cognitive modeling and evaluation of Human-Machine Interfaces.

Dr. Hélène Pigot is professor in computer science at Sherbrooke University, Canada. She co-founded the DOMUS laboratory. She is director of the Center on smart environment of the Sherbrooke University. Hélène Pigot received his PhD on speech recognition in 1985, at P.M. Curie University, France. She was then graduated in occupational therapy at Montreal University, Canada. Professor at Montreal University during five years, she worked on spatial orientation deficits in Alzheimer disease. Since 2000, she dedicates her research on smart home, cognitive modeling and cognitive assistance for people with cognitive deficits.