International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:12, 2008

Analytical Model Based Evaluation of Human Machine Interfaces Using Cognitive Modeling

Belkacem Chikhaoui, Hélène Pigot

***Abstract*—**Cognitive models allow predicting some aspects of utility and usability of human machine interfaces (HMI), and simulating the interaction with these interfaces. The action of predicting is based on a task analysis, which investigates 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 system's functionalities. Cognitive models are part of the analytical approaches, that do not associate the users during the development process of the interface. This article presents a study about the evaluation of a human machine interaction with a contextual assistant's interface using ACT-R and GOMS cognitive models. The present work shows how these techniques may be applied in the evaluation of HMI, design and research by emphasizing firstly the task analysis and secondly the time execution of the task. In order to validate and support our results, an experimental study of user performance is conducted at the DOMUS laboratory, during the interaction with the contextual assistant's interface. The results of our models show that the GOMS and ACT-R models give good and excellent predictions respectively of users performance at the task level, as well as the object level. Therefore, the simulated results are very close to the results obtained in the experimental study.

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

I. Introduction

THE evaluation of human machine interfaces is becoming increasingly important. While their development presents some challenges, the evaluation of interfaces needs rigorous methods to ensure that they fulfill the initial specifications as well as the quality of accessibility and the usability of these interfaces [1], [2].

Two main approaches are currently used for the evaluation of HMI. The first one is empirical approaches, which are essentially based on performances or opinions of users gathered in laboratories or other experimental situations. The second one is analytical approaches, which are not based directly on the user performance, but on the interfaces' examination using well defined structures and rigorous analytical techniques [3]. Analytical approaches allow to predict mainly user performance, time execution of tasks, performance design and the explanation of an existing interface's performance [4]. Since these approaches can predict time execution of tasks, this latter should be accurately measured and evaluated. This can be done by adding each time the user interacts physically with the interface, either by the stroking on a keyboard, pointing with the mouse on the screen, or even by pointing with a human finger on a touch screen.

Paul M Fitts has defined the law of physical components of the interaction with the interface by making the analogy between the interface and the target to reach [5]. However, reaching physically a component of the interface supposes that, a cognitive process was engaged before choosing the component to interact with. Therefore, the interaction process with the interface implies three human components. The first component is perceptual, which concerns more specifically the visual and aural perceptions in HMI. The second component is cognitive, implying the human to reason and retrieve in his memory the application of rules and the remembrance of objects in order to satisfy specific goals [6]. The third and the last component is motor, where the user reaches and interacts with the specific interface component.

The most important challenge of analytical methods is their capability to define and simulate the three components elicited in the HMI, in order to predict the users behavior.

The action of prediction can be performed using predictive models, which are an integral part of analytical approaches.

The GOMS is a predictive model, which estimates the time a user interacts with the interface, taking in account the time requested for the cognitive process to select the appropriate interaction [7]. ACT-R, a cognitive architecture, predicts the time needed to perceive stimulus, either aural or visual, to retrieve knowledge in memory and to execute the motor actions [6]. The two methods give opportunity to explain the way users accomplish goals.

In this study, we aim to evaluate the interaction with the interface of a contextual assistant application, developed to help persons with cognitive disabilities perform autonomously their daily living tasks. This application assists people while preparing meals in the kitchen by using cognitive assistance [8]. Due to the related population and the kind of errors they commit, we need to take in account the cognitive part involved in the interaction with the HMI. Then we use a powerful analytical methods based specifically on cognitive models to evaluate the contextual assistant's interface, emphasizing the cognitive analysis of the tasks in one side, and the time execution of these tasks on the other side.

Our analytical evaluation is based on two methods. The first method simulates the task thanks to the cognitive architecture ACT-R [6] in which the interaction is decomposed in rules simulating the behavior of a human interacting with contextual assistant's interface. The second method is the GOMS model (Goals, Operators, Methods and Selection rules), which is a formalized representation that can be used to predict task

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

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

performance [9]. The GOMS model is a way in which users achieve goals by solving subgoals in a divide-and-conquer fashion [10].
In order to create an effective evaluation, an empirical study is conducted at the DOMUS laboratory over ten healthy persons. The results of our models are compared with those obtained in the experimental study.
After introducing a theoretical background about the concept of evaluation (section II), we present an overview of the analytical methods chosen to evaluate the contextual assistant's interface: the cognitive architecture ACT-R and the GOMS model (section III). The interface to be evaluated is presented in (section IV) and the experimental study is then introduced in (section V). The models developed are then presented in (section VI) and the results of the simulation are compared to the results obtained in the experimental study (section VII).

II. Theoretical Background

The evaluation of systems focuses on two main aspects : the utility and the usability. The utility is defined as the question of whether the functionality of the system can do what is needed [11], and the usability is defined as the easiness of learning and using the system [12]. The evaluation of HMI ensures that the applications fulfill the users needs and requirements, and ensures that the interaction is motivate and enjoyable. Due to the usability problems detected during the evaluation process, more efficiency, adaptability and accessibility are expected in the interface [13], [14].
According to J. Preece and al, the evaluation can be defined as " the process of systematically collecting data that informs us about what it is like for a particular user or group of users to use a product for a particular task in a certain type of environment" [12].
The evaluation is either empirical or analytical depending on the used methods. While the empirical methods evaluate the performance of an interface when users interact with it, the analytical methods simulate a user behavior based on theoretical knowledge.
The empirical methods are widely used in the literature to evaluate interfaces in various situations, either for traditional interaction with computers [15], [16], [17], mobile devices [18], [19] or pervasive interfaces [20]. Caution is needed for these methods to ensure that the subjects selected for the experimentation are representatives of the final users. During the test, the tasks to perform need also to be carefully designed to evaluate the way the final users will interact with the application. The evaluation of the interactions in real settings with a numerous set of people constitutes the trends of empirical evaluation. For example, the evaluation of cell phone menu's interaction requires fourteen experienced cell phone users performing tasks [18].

Pervasive computing systems involve different systems that make the process of evaluation difficult. The necessity to evaluate interfaces in real settings leads to long experiments when pervasive computing is evaluated at home, for instance three month period test by a young couple living in a smart apartment were needed to evaluate pervasive computing system at home [19], [20]. Thus, the empirical evaluation necessitates high costs and time consuming.
To mitigate these drawbacks, analytical evaluation allows to simulate as much users as needed to perform various tasks on different versions of the interface. The analytical evaluation of HMI is based on theories and methods and the results bring a clear understanding of the way the users interact with the interface [21]. Analytical methods predict and identify practical errors and usability problems [14]. The analytical evaluation process should be conducted to evaluate the applications as well as the HMI. P Antunes and al propose to evaluate the groupware design (collaborative tool), which is a multi-user context using an analytical method derived from the GOMS model [22]. The tasks, users and the environment are then modeled. Therefore, different situations are simulated varying upon the different versions of the interface, the tasks to perform and the abilities of the user. According to St. Amant and al, the analytical evaluation of the smart phone menu's interaction increases the optimum version of menu on small screens [18]. The cognition evaluation should also demonstrate how the time is shared between the three components involved during the interaction [23]. Through ACT-R model, D. Salvucci demonstrates the impact of phone call during driving [24].

III. Analytical methods to evaluate the contextual assistant's interface

The contextual assistant application aims to help people to complete daily living's activities. It is dedicated for people with cognitive deficits to foster autonomy, such as people with mental retardation. These users fail due to difficulties in planing, memory and attention. The interface must be specially designed for this population. However, the involvement of this kind of population during the evaluation process must remain limited. To do so, the researchers develop analytical methods to avoid numerous empirical evaluation problems as mentioned previously.
Due to the related population, the analytical approaches should emphasize the cognitive and perceptual processes required while using the contextual assistant. Therefore, we choose analytical methods based on cognitive theories, which are GOMS and ACT-R. Those methods are derived respectively from the human model processor theory and from the unified theories inspired by the work of Allen Newell [25], [26]. The objective of the study is to validate the cognitive simulation using these two methods and to analyze how they provide theoretical explanations upon the errors committed by the users. The two next sections present the analytical methods used in the simulation.

A. Cognitive Architecture ACT-R

The cognitive architecture ACT-R is built to simulate and understand human cognition [6], [27]. It consists of a set of modules such as the visual, aural, motor, intentional and declarative module that are integrated through a central production system. ACT-R is an hybrid architecture that combines two subsystems: symbolic system including semantic and

procedural knowledge, and subsymbolic system evaluating knowledge activations.

Each knowledge in the ACT-R's declarative memory is called chunk, and is associated with a level of activation computed by the subsymbolic system [26]. The activation level reflects the degree of availability of the chunk at any particular time. The subsymbolic system assigns also utility values to rules (procedural knowledge) to determine the predominant knowledge available at a specific time. The predominant knowledge is defined as the rule with the highest utility.

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 [28], [29].

The visual module is decomposed in two subsystems, the positional system (where) and the identification system (what), that work together in order to send the visual stimulus to the visual buffer. The positional system is used to find objects. When a new object is detected, the chunk that represents 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 represents a visual location that 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 [28], [29].

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 is an acronym for Goals, Operators, Methods and Selection rules. It is a formalized method used to predict task performance [7], [9], [10]. A GOMS description consists of these 4 elements:

1) Goals: The user's goals describe 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.

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.

Each task is decomposed hierarchically in goals and subgoals according to the divide and conquer technique. The subgoals are also decomposed down until reaching the basic operations description. The total execution time is then estimated by summing the times of basic operations.

IV. Contextual Assistant

After having presented the tow analytical methods selected to conduct our evaluation, we present now the application to be evaluated.

The Contextual assistant is an application developed to assist persons with cognitive disabilities [30], [31]. The aim is to foster autonomy in daily living tasks, and particularly during complex cooking tasks such as preparing spaghetti [8]. The cooking task is decomposed of steps that are displayed on a touch screen. The two first steps consist of gathering the utensils and ingredients necessary to the recipe (Fig. 1). The other steps explicit the recipe using photo and video on the screen and also explicit the 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 that displays the objects to search. When an object is selected on the main interface, the contextual assistant looks for the location of that object in the environment using techniques of pervasive computing, and indicates the object 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 second to use the object locator in order to locate each object in the environment.

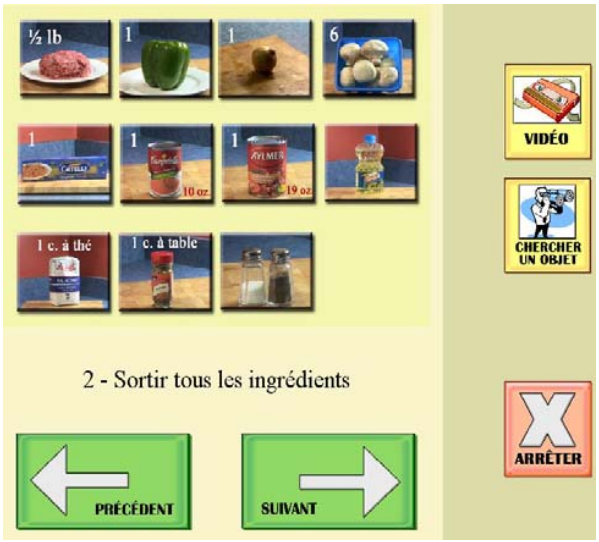

Fig. 1. The contextual assistant's interface representing the gathering ingredients task

The contextual assistant's 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.

V. Experimental Study

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

A. Apparatus and Application

The experiment consists on selecting items in the contextual assistant's interface. The items correspond to the list of utensils and ingredients needed to realize the cooking task. Each item is displayed with a large button in the contextual assistant's interface (Fig. 1). The experiment is conducted according to three main criteria:

1) The time to select items is measured accurately.
2) The order in which the objects are displayed does not affect the speed of selection.
3) The experiment is uniform for all subjects.

To ensure the first criterion, we decide to isolate the time of object recognition and the time of selection in the contextual assistant's interface. Therefore, we measure the time needed to decide which object to get out and we measure the time to push on that object in the interface.

The first action corresponding on deciding which object users want to get out is presented experimentally using a PDA (Personal Digital Assistant). The name of the object to get out is displayed on the PDA in order to highlight the phase of objects recognition involved in the cognitive processes. To avoid automatic selection on the contextual assistant's interface, the names of objects displayed on the PDA are chosen randomly.

Knowing the object name, the subject executes the second phase which consists of pushing the correspondent button on the contextual assistant's interface. For each object needed in the experiment, the two phases' times are recorded in a log file and recovered at the end of experiments.

B. Subjects

Ten students of Sherbrooke's University participate to the study. All subjects are male and their ages range from 27 to 32 years. The subjects have good vision with no physical impairments being reported. All subjects have a good knowledge in computer science, but they have no prior knowledge in the application and the cognitive assistance field.

C. Method

The PDA is placed at a distance of 15 cm from the touch screen, subjects remain standing at a distance of approximately 30 cm from the touchscreen during the entire test as shown in Fig. 2.

The subjects familiarize themselves with the interface during a practical stage. When the test begins, the subjects look first on the PDA to know the name of the object to get out and second, push the corresponding button in the contextual assistant's interface using the index finger. To know the next object to reach, the subjects click on the PDA to display its name.

The experiment continues until the last object of the gathering objects' task is reached. The objects displayed on the PDA are presented to subjects under a random order. This emphasizes the recognition of object's phase in the cognitive process. Each subject accomplishes 5 trials, where a trial is composed of two tasks, which are gathering utensils and gathering ingredients. Each trial needs 25 actions "pressing on the PDA" and 25 actions "pushing button in the contextual assistant's interface". Altogether 2500 (10 subjects x 5 trials x (25 actions x 2 interfaces) = 2500) actions are observed during the experiment.

Fig. 2. The human machine interaction during the experimental study

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 in selecting each object in the two tasks, over all subjects in our study.

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

Objects	Duration (s)	Standard Deviation (S)
LOOK-FOR-OBJECT (1)	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 (1)	2.039	0,386
NEXT	2.142	0,540
LOOK-FOR-OBJECT (2)	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 (2)	2.432	0,614

VI. Modeling The Interaction With The Contextual Assistant

In this section, we present the modeling process of the tasks evaluated in our study, which are gathering utensils and gathering ingredients. After analyzing the tasks to modelize, we present first the model with ACT-R and second the model with GOMS.

A. Task analysis: gathering utensils and ingredients

The two first steps of the recipe gathering utensils and ingredients require three subtasks (Fig. 3). The first subtask consists of activating the object locator in order to locate each needed object for the recipe. This is done by pushing the button "LOOK-FOR-OBJECT (2)", which is displayed in the main contextual assistant's interface (Fig. 1). The second subtask consists of locating 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 contextual assistant's interface in order to know the next step of the recipe. The tree decomposition of the gathering ingredients' task is presented in Fig. 3. The nodes in capital indicate the action to click on the named button, while the other nodes represent the tasks to be decomposed.

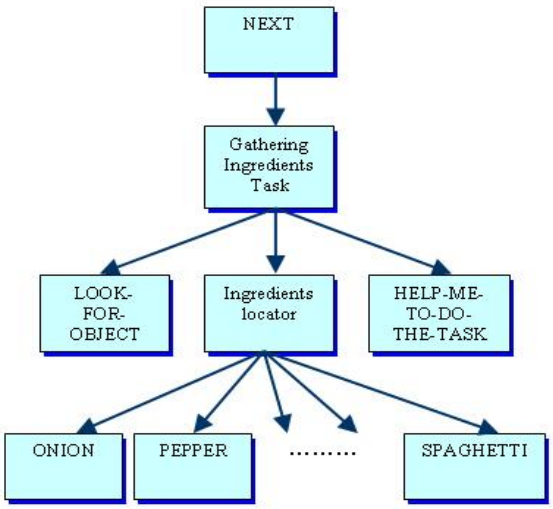

Fig. 3. Tree representation of the gathering ingredients task

B. Modeling The Interaction With The Contextual Assistant using ACT-R

The model uses ACT-R to emphasize the cognitive processes involved, when looking on an object and when choosing the button to push. The model is subdivided in three phases, the visual phase, the recognition phase and the motor phase.
The visual phase consists of two steps: localizing the object to perceive and 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 object is the button "LOOK-FOR-OBJECT (2)" as described in Fig. 3. Then, all the utensils (or ingredients) needed for 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 (2)" is presented in order to come back to the main contextual assistant's interface 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 made request to the visual-location buffer of ACT-R, which creates a chunk representing the location of the specified object. When the location' step is over, the identification system identifies the name of the object and creates a chunk. This chunk is placed in the visual buffer. The steps of location and identification last 185 (ms) [28], [29]. 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 recognition's phase begins when the chunk of the object is stored 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's phase consists of activating the motor actions through 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 ingredients' task is reached.

In our ACT-R model, the contextual assistant's interface 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 Fig. 4.

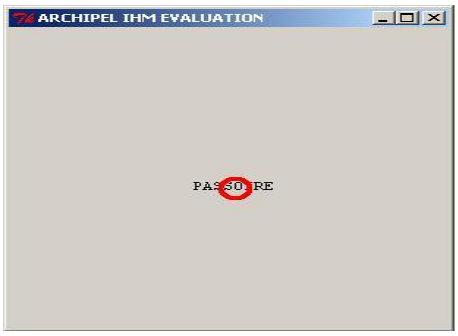

Fig. 4. Shift attention representation in the ACT-R model

The ACT-R model is developed using the ACT-R 6 environment. All memory chunks get the same value of activation and all requests to the retrieval buffer are correctly satisfied without errors. These characteristics lead to a deterministic model where no error could happen.
Fig. 5 shows in a very low detailed form an example of the execution trace of the ACT-R model. The visual phase is

principally based on the shift attention and the visual encoding actions, which are presented in the section I of Fig. 5 followed by the recognition phase in section II of Fig. 5. Finally, the motor phase is presented in section III of Fig. 5. 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 still needs 0.085 (seconds) to be completed and stores the chunk into the visual buffer. During the recognition 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). Finally a request on the motor buffer starts at time 0.447 (seconds).
In the experimental study, users interact with the PDA and the screen. Each device involves three cognitive processes including the visual, cognitive and motor phase. Therefore, the ACT-R model simulates the time twice during the interaction with each object, on the PDA firstly and on the touch screen secondly. The simulation time is computed as the summation of the time estimated for each object to click on the PDA and on the touch screen.

```
Section 1. Visual Phase

0.000     GOAL SET-BUFFER-CHUNK GOAL FIRST-GOAL REQUESTED NIL
0.000     PROCEDURAL     CONFLICT-RESOLUTION
0.050     PROCEDURAL     PRODUCTION-FIRED START-APPLICATION
THE SUBJECT STARTS TO LOOK FOR NEW OBJECT
0.050     PROCEDURAL     CLEAR-BUFFER IMAGINAL
0.050     PROCEDURAL     CLEAR-BUFFER VISUAL-LOCATION
0.050     PROCEDURAL     CLEAR-BUFFER GOAL
0.050     VISION         Find-location
0.050     VISION         SET-BUFFER-CHUNK VISUAL-LOCATION LOC1
0.050     GOAL           SET-BUFFER-CHUNK GOAL GET-OBJECT0
0.050     PROCEDURAL     CONFLICT-RESOLUTION
0.100     PROCEDURAL     PRODUCTION-FIRED ATTEND-UTENSIL
SHIFT ATTENTION TO A SPECIFIED LOCATION ON THE SCREEN
0.100     PROCEDURAL     CLEAR-BUFFER VISUAL-LOCATION
0.100     PROCEDURAL     CLEAR-BUFFER VISUAL
0.100     PROCEDURAL     CONFLICT-RESOLUTION
0.185     VISION         Encoding-complete LOC1-0 NIL
0.185     VISION         SET-BUFFER-CHUNK VISUAL TEXT1
0.185     PROCEDURAL     CONFLICT-RESOLUTION

Section 2. Recognition Phase

0.235     PROCEDURAL     PRODUCTION-FIRED ENCODE-UTENSIL
ENCODING THE OBJECT AFTER VISUAL ATTENTION
0.235     PROCEDURAL     CLEAR-BUFFER VISUAL
0.235     PROCEDURAL     CLEAR-BUFFER IMAGINAL
0.235     IMAGINAL       SET-BUFFER-CHUNK IMAGINAL OBJECT1
0.235     PROCEDURAL     CONFLICT-RESOLUTION
0.285     PROCEDURAL     PRODUCTION-FIRED FOUND-OBJECT
RETRIEVE THE CORRESPONDING CHUNK FROM THE DECLARATIVE MEMORY
0.285     PROCEDURAL     CLEAR-BUFFER IMAGINAL
0.285     PROCEDURAL     CLEAR-BUFFER RETRIEVAL
0.285     DECLARATIVE    START-RETRIEVAL
0.285     PROCEDURAL     CONFLICT-RESOLUTION
0.397     DECLARATIVE    RETRIEVED-CHUNK OBJECT1-0
0.397     DECLARATIVE    SET-BUFFER-CHUNK RETRIEVAL OBJECT1-0
0.397     PROCEDURAL     CONFLICT-RESOLUTION

Section 3. Motor Phase

0.447     PROCEDURAL     PRODUCTION-FIRED MOTOR-ACTION
THE SUBJECT PUSHS THE CORRESPONDING ICON OF THE OBJECT
0.447     PROCEDURAL     CLEAR-BUFFER VISUAL
0.447     PROCEDURAL     CLEAR-BUFFER RETRIEVAL
0.447     PROCEDURAL     CLEAR-BUFFER MANUAL
```

Fig. 5. Example of execution trace of the ACT-R model

C. Modeling The Interaction With The Contextual Assistant Using GOMS

The first two steps of the recipe, gathering utensils and gathering ingredients have been described previously, which can be interpreted in the GOMS language by a method that is divided in three steps as shown in Fig. 6. Each step defines a new goal to be reach.

```
Method_for_goal: Archipel Evaluation

        Step 1. Accomplish_goal: Select Utensils.

        Step 2. Accomplish_goal: Select Ingredients.

        Step 3. Return_with_goal_accomplished.
```

Fig. 6. Main method of the GOMS model

For each step in our study, a method is defined according to 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, by solving subgoals the user achieves goals until reaching the basic operations called "operators", which can not be subdivided [10]. The methods have a hierarchical structure. Therefore, a method may call for subgoals to be accomplished [32]. Fig. 7 shows explicitly the subgoal's invocations in the hierarchy of the "Select Ingredients" subgoal.

```
Method_for_goal: Select Ingredients

 Step 1.  Accomplish_goal: Select NEXT.
 Step 2.  Accomplish_goal: Select LOOK-FOR-OBJECT(2).
 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(2).
 Step 15. Return_with_goal_accomplished.
```

Fig. 7. GOMS Method for Select Ingredients task

The main method presented in Fig. 6 constitutes the root of the tree hierarchy and all the other methods are generated automatically using the divide-and-conquer technique [10]. In our GOMS model, each object is defined as visual object. The selected methods have the same form for all objects. Fig. 8 explains the tree decomposition corresponding to the main method of the GOMS model.

The duration of a step in the GOMS model can be defined as the sum of the production cycle's duration and the duration of all actions included inside the step. Therefore, the production cycle's duration equals to 50 (ms) and for instance, the performance of key presses is estimated to 280 (ms) [33], [18].
Our GOMS model is executed using the GLEAN3 modeling tool [33].

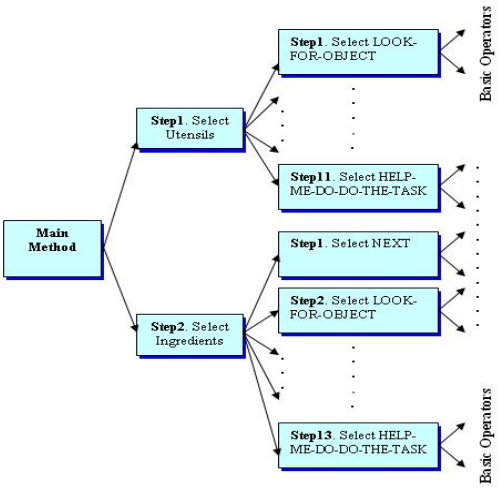

Fig. 8. Tree decomposition of the main method in the GOMS model

VII. Comparison of Results

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

A. Object level performance

Table II shows the comparison of the time needed by the user, the ACT-R and GOMS model predictions. Values in parentheses represent the smallest and greatest time needed by the user to press each object. The objects "LOOK-FOR-OBJECT (1)" and "LOOK-FOR-OBJECT (2)" are displayed respectively on the gathering utensils and gathering ingredients interfaces and the same thing is applied for the objects "HELP-ME-TO-DO-THE-TASK (1)" and "HELP-ME-TO-DO-THE-TASK (2)". Fig. 9 and Fig. 10 show respectively the predicted time of each object during the gathering utensils and gathering ingredients tasks in a detailed graphical form. According to Fig. 9 and Fig. 10, the results of both ACT-R and GOMS models are very close and have approximately the same predicted time values for several objects.

B. Task level performance

Table III shows the user performance data, the ACT-R and GOMS model predictions in both tasks: gathering utensils and gathering ingredients. Fig. 11 shows the same data in a detailed graphical form.

Fig. 12 shows the progression in performing tasks over the time in the experimental study, ACT-R and GOMS models. Since the prediction's procedure is applied for each object in the interface, the two models follow a linear model. This is supported by some scientific literature [18]. 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 Fig. 12.

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 (1)	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 (1)	2.039 (1.349 2.680)	2.165	2.550
NEXT	2.142 (1.289 3.082)	2.165	2.550
LOOK-FOR-OBJECT (2)	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 (2)	2.432 (1.265 3.654)	2.165	2.550

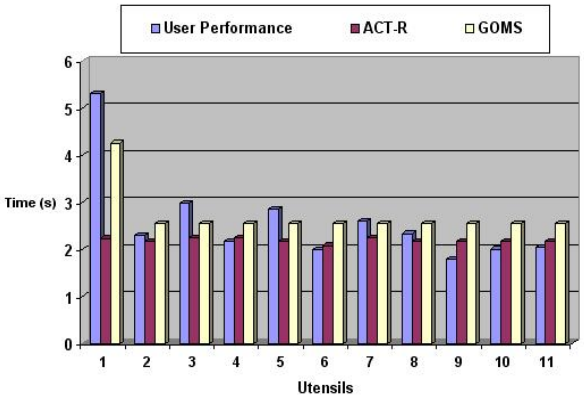

Fig. 9. User data, ACT-R and GOMS model predictions for the gathering utensils task

VIII. General Discussion

The ACT-R and GOMS models, which 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 excellent predictions of time execution of tasks as well as objects as shown respectively in Table II and Table III.

As shown in Table II, the object "LOOK-FOR-OBJECT (1)" needs more time to be pushed using the GOMS model

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 (11 objects)	28.290	24.050	29.750
Gathering ingredients (14 objects)	28.973	30.650	35.800

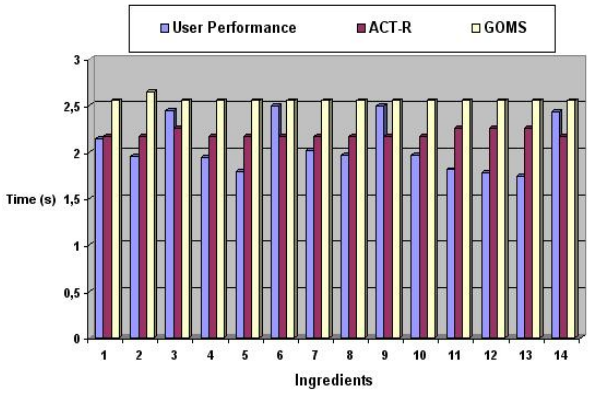

Fig. 10. User data, ACT-R and GOMS model predictions for the gathering ingredients task

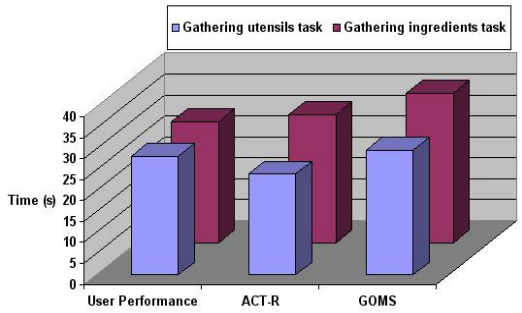

Fig. 11. User data, ACT-R and GOMS model predictions by task

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 gathering utensils task. This mental operator takes more than one second to be accomplished. The same difference is observed in the predicted time of the object "LOOK-FOR-OBJECT (2)" 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.

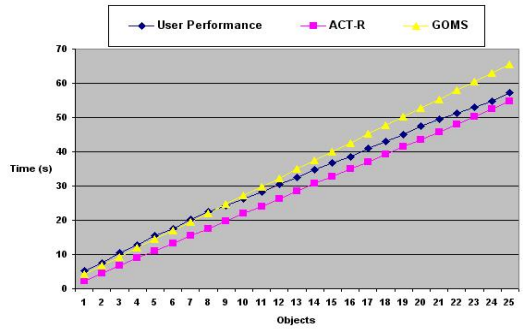

Fig. 12. Progression in accomplishing tasks over the time

Some differences in the predicted time of some objects using the ACT-R model are observed 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 to 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 fact that the objects are not displayed in the same location on the screen, but subjects in the experimental study performed differently depending on the exact position of the objects on the screen. For instance, the object "PEPPER", which is displayed at the top of the screen as shown in Fig. 1, needs more time to be selected 2.448 (seconds) than the object "TOMATOES-BOX", which is displayed in the center of the screen and needs 1.794 (seconds) to be selected.

In Table III, an important remark must be mentioned. The time taken to accomplish the first task in the experimental study, which equals to 28.290 (seconds) is very close to the time taken to accomplish the second task, which equals to 28.973 (seconds). Although the number of objects needed for the first task (11 objects) is lower than the one needed for the second task (14 objects). Two interpretations are possible:

1) The first interpretation is that the user learns gradually the position of objects in the interface and the navigation in the interface. Due to the learning, the second task will be performed faster than the first one and the last experiments will be performed generally faster than the first experiments.
2) The second interpretation is related to the nature of objects displayed in the contextual assistant's interface. At each name given on the PDA corresponds an image on the contextual assistant's interface. Some objects of the first task seem to be more difficult to identify such as the one shown in Fig. 13. This should provoke delay in the execution of the tasks.

Fig. 13. Example of an object not easily identifiable

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.

IX. Conclusion

This study empirically demonstrated that cognitive models are a powerful tool for evaluating interfaces and predicting user's performance. The main goal of our study is to build and validate models for the evaluation of the contextual assistant's interface by simulating the HMI 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. 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. The results show that the GOMS model can predict user's performance at good level and the ACT-R model can predict user's performance at more detailed level and performs almost as well. Our models are powerful and realistic as demonstrated with the comparison of the time taken by subjects performing the same tasks.

According to these results, the two models could be used to improve the design of the contextual assistant's interface and to optimize it.

During the conception of the GOMS and ACT-R models, we observed that the GOMS model gives more flexibility in modeling than the ACT-R model, which constitutes the complicated part in our study. However, the ACT-R model proposes a more accurate explanation about the cognitive processes involved during the interaction with the contextual assistant's interface, and hence, the possibility of introducing cognitive errors.

Our study makes two main contributions, the first contribution is to design an analytical evaluation of HMI designed for cognitively impaired people. It constitutes a new study in that field. The second contribution is the use of cognitive models to evaluate these interfaces emphasizing on the cognitive processes involved during the human machine interaction.

X. Future Improvements

Some improvements should be brought to our models. First, our models are deterministic and do not make errors. They should be extended to allow errors in the pointing actions such as pushing an object several times before or after looking 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 and during the interaction with the contextual assistant's interface [20], [34], [35].

Second, since the contextual assistant is designed to assist cognitively impaired people in smart homes, it would be interesting to do some experiments with this population. The comparison between the results of a non deterministic model and the experiments' results 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 summarized into the human machine interaction 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] 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.

[4] 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.

[5] P. M. Fitts, "The information capacity of the human motor system in controlling the amplitude of movement," *Journal of Experimental Psychology*, vol. 47, NO. 6, pp. 381–391, 1954.

[6] 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.

[7] 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, pp. 287–319, 1996.

[8] 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.

[9] 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.

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

[11] J. Nielsen, *Usability Engineering*. Boston, MA: Academic Press, 1993.

[12] J. Preece, Y. Rogers, and H. Sharp, *Introducing evaluation*, ser. Interaction Design: Beyond Human-Computer Interaction. John Wiley and Sons, 2002, ch. 10, pp. 317–337.

[13] J. Nielsen and R. Molich, "Heuristic evaluation of user interfaces," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM New York, NY, USA, 1990, pp. 249–256.

[14] L.-O. Bligard and A.-L. Osvalder, *An Analytical Approach for Predicting and Identifying Use Error and Usability Problem*, a. holzinger. ed., ser. HCI and Usability for Medicine and Health Care, 2007, vol. 4799/2007, ch. 38, pp. 427–440.

[15] C. Stephanidis, D. Akoumianakis, M. Sfyrakis, and A. Paramythis, "Universal accessibility in hci: Process-oriented design guidelines and tool requirements," in *Proceedings of the 4th ERCIM Workshop on User Interfaces for All*, 1998, pp. 1–15.

[16] A. C. Siochi and D. Hix, "A study of computer-supported user interface evaluation using maximal repeating pattern analysis," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1991, pp. 301–305.

[17] J. H. Goldberg and X. P. Kotval, "Computer interface evaluation using eye movements: methods and constructs," *International Journal of Industrial Ergonomics*, vol. 24, no. 6, pp. 631–645, 1999.

[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] T. koskela, K. V.-V. Mattila, and L. Lehti, "Home is where your phone is: Usability evaluation of mobile phone ui for a smart home," in *Proceedings of MobileHCI 2004*, vol. 3160, 2004, pp. 74–85.

[20] B. E. John and D. D. Salvucci, "Multi-purpose prototypes for assessing user interfaces in pervasive computing systems," *IEEE pervasive computing*, vol. 4, no4, pp. 27–34, 2005.

[21] B. Chikhaoui and H. Pigot, "Simulation of a human machine interaction: Locate objects using a contextual assistant," in *procedeengs of the 1st International North American Simulation Technology Conference*, M. Beldjehem, Ed., 2008, pp. 75–80.

[22] P. Antunes, M. R. S. Borges, J. A. Pino, and L. Carrio, *Analytic Evaluation of Groupware Design*, ser. Computer Supported Cooperative Work in Design II, 2006, vol. 3865/2006, ch. 4, pp. 31–40.

[23] B. Chikhaoui and H. Pigot, "Evaluation of a contextual assistant interface using cognitive models," in *procedeengs of the 5th International Conference on Human-Computer Interaction*, waset, Ed., vol. 34, 2008, pp. 36–43.

[24] D. D. Salvucci and K. L. Macuga, "Predicting the effects of cellular-phone dialing on driver performance," *Cognitive Systems Research*, vol. 3, no. 1, pp. 95–102, 2002.

[25] B. E. John and D. E. Kieras, "The goms family of analysis techniques: Tools for design and evaluation," Carnegie Mellon University School of Computer Science, Technical Report CMU-CS-94-181, 1994.
[26] A. Newell, *Unified Theories of Cognition*. Cambridge (Mass): Harvard University Press, 1990.
[27] 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.
[28] 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.
[29] D. Bothell, "Act-r 6.0 reference manual," Working Draft, 2004.
[30] 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.
[31] D. Lussier-Desrochers, Y. Lachapelle, H. Pigot, and J. Bauchet, "Apartments for people with intellectual disability: Promoting innovative community living services," in *Proceedings of the 2nd International Conference on Intellectual Disabilities/Mental Retardation.*, 2007.
[32] 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, pp. 320–351, 1996.
[33] 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.
[34] 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.
[35] A. Dion and H. Pigot, "Modeling cognitive errors in the realization of an activity of the everyday life," in *Cognitio*, 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 University of Sherbrooke. Sherbrooke, QC, Canada. His research interests include smart homes, 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.