Controlling Biases in Demanding Tasks

Annerieke Heuvelink[1, 2] (heuvel@few.vu.nl) and Jan Treur[2] (treur@few.vu.nl)

[1] TNO Defense, Security, and Safety, Training and Instruction, P.O. Box 23, 3769 ZG Soesterberg, The Netherlands
[2] Vrije Universiteit Amsterdam, Artificial Intelligence, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

Abstract

Many aspects affect the way humans perform tasks: not only the content of a task, but also somebody's personality or his or her current exhaustion. Under varying conditions the quality of the performance is known to vary, for example, due to biases that occur. This paper introduces a cognitive model addressing these aspects. It has been formally specified, tested in simulations for various scenarios, and formally analysed.

Keywords: Modeling; Control; Biases; Task Performance.

Introduction

Humans differ in how they perform a task, given aspects of environmental pressure and the person's characteristics and internal states. Variability in task performance may affect the quality of performance. It is well-known that stress or high task demands can deteriorate task performance. At the basis of these findings lies the fact that humans have a limited amount of cognitive resources. When a task becomes more demanding, these resources might become insufficient. To deal with this, humans tend to apply cheaper cognitive reasoning steps such as heuristics, while performing the task. These shortcuts often work well and might even be regarded as adaptive given their ecological validity (Gigerenzer et al., 1999). However, when the outcome of such a reasoning step deviates in a structural way from the rational outcome, it is called a bias.

The challenge addressed in this paper is to design a computational model for task performance that addresses how biases and their control relate to internal and external factors as mentioned. Various applications may benefit from such a model of human-like task performance. For example, it can be used to design virtual characters that play a role in simulations in which human aspects are important, such as in realistic training environments and social games. Furthermore, such a model may also help a software agent to better understand human behavior in cooperative task performance and thus aid decision support.

In this paper first some background literature considering human task performance is briefly summarized. Next, a control approach is introduced capable of generating variable task behavior. After this it is shown how this approach to control can be used to address the control of biases in task performance, resulting in a formally specified cognitive agent model. Moreover, it is described how this model was tested in a case study in which the agent operates under a variety of scenarios involving different internal and external conditions. Finally, the model is evaluated, it is discussed how this work relates to other work, and future research directions are exposed.

Human Task Performance

Individual differences in cognitive characteristics and states entail differences in human task performance. In general some humans are more gregarious, impulsive, distractible, and less patient than others (Shields, 1983). At the same time humans manage the limited resources they have in certain ways, see e.g., (Johnston & Heinz, 1978). The allocation of cognitive resources is claimed to be flexible and under own control.

Kahneman (1973) introduced the idea of humans having limited cognitive resources. He states that there is no exact fixed amount, but that that is influenced by the arousal level of a person: the higher the arousal level, the more resources can be made available, up to a certain point. From that point on an increase in arousal may not result in an increase of available resources. McBride et al. (2007) reaffirm this and also point out that humans are able to perform multiple tasks at once, as long as the total sum of processing demands does not exceed the available resources. When the total sum does exceed this level of available resources, task performance will decline (Posner & Boies, 1971).

A method humans apply to bring down the cognitive demands of a task is the use of heuristics. These rules of thumb often work in certain types of situations. Characteristics of heuristics besides context dependency are their simplicity and speed. However, when they deliver incorrect or inaccurate results, they are also referred to as biases, for an overview see (Wickens & Flagg, 1988). Cognitive biases are known to arise especially under stress of overload conditions (Baron, 2000) and have an immediate impact on the quality of task performance.

Hancock and Warm (1989) acknowledge that demanding tasks over time do, through some kind of physiologically mediator, influence cognition. They forward the thesis that tasks themselves are the major sources of cognitive stress, which others support (e.g., Matthews & Desmond, 2002).

Model Setup and Control Approach

To mimic the variety in human task performance, the model possesses multiple cognitive processing components that can generate the same outcome. These components vary in content, so the model can display the variety found in task performance. Some are rational and generate the output in a correct way, others represent typical biases and 'forget' to take certain factors into account. Components that perform biased processing require less processing resources, but they may generate incorrect outputs.

Moreover, the model possesses a control method to determine which of the cognitive processing components may become active to generate a required output, see Fig. 1. On the top level, above the dashed line, the control processes are shown, distinguished from the component processing. Input for this control process is coordination information about the various components and their input-output connections. The output of this control level is control information on which components should become active. Each component has two input layers: one for coordination information (the upper square at the left side of the component), and one for data information (the lower square). Output is also generated at both levels, depicted by the squares at the right side of a component.

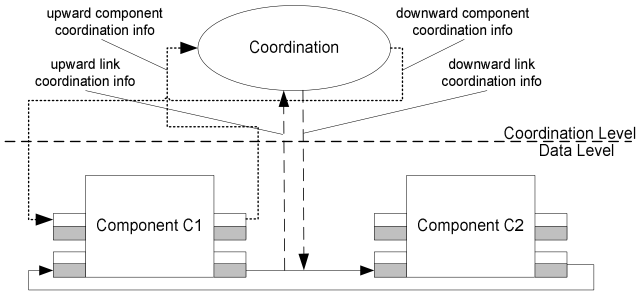

Figure 1: Control Approach

The model decides which component(s) may become active based on the current external as well as internal states. A major constraint is that the required processing resources of the to-be-selected components have to lie within the available processing resources.

As discussed above, cognitive biases arise in human task performance under overload conditions. Since the cognitive agent model is to mimic human (biased) task performance, the same principle should hold for an agent incorporating the model. The idea is that when faced with high task demands the agent will be motivated to operate on a high cognitive processing level. Over time this will result in it becoming exhausted, which entails less available processing resources, which will affect the control decisions made. Specifically, when the agent becomes more exhausted, components with lower processing costs will be chosen, which usually implies a higher level of biases.

Formal Analysis

The cognitive model is expected to show certain behavioral properties as discussed above. Here such properties are identified and formalized enabling automated verification. The first property relates task demand to biases.

HTDtoHB Higher task demand leads to higher biases.

This global property can be related to more local properties relating task demand to exhaustion, exhaustion to selection of less demanding components, and less demanding components to biases.

HTDtoHX Higher task demand leads to a higher exhaustion level.

HXtoLDC Higher exhaustion level leads to less demanding components.

LDCtoHB Less demanding components lead to higher biases.

The relationship between these behavioral properties is:

HTDtoHX & HXtoLDC & LDCtoHB ⇒ HTDtoHB

For formalization of these properties a reified temporal predicate logical language was used; e.g. (Galton, 2006). Expressions are built on atoms referring to state properties, time points and traces. The properties can be formalised by comparing for one given trace the levels (of task and component demand, exhaustion level and biases) to certain bounds, or by comparing these levels in a relative manner in two traces. The following abbreviations are used.

aboveduring(γ, t, D, a(V), M) ≡ ∀t1,V1 [t≤t1≤t+D & at(γ, t1, a(V1)) ⇒V1≥M
belowduring(γ, t, D, a(V), M) ≡ ∀t1,V1 [t≤t1≤t+D & at(γ, t1, a(V1)) ⇒V1≤M
aboveleadstoabove(γ,D1, a(V), M1, E, D2, b(V), M2) ≡
∀t [aboveduring(γ, t, D1, a(V), M1) ⇒ aboveduring(γ, t+E, D2, b(V), M2)]
aboveleadstobelow(γ,D1, a(V), M1, E, D2, b(V), M2) ≡
∀t [aboveduring(γ, t, D1, a(V), M1) ⇒ belowduring(γ, t+E, D2, b(V), M2)]
belowleadstoabove(γ,D1, a(V), M1, E, D2, b(V), M2) ≡
∀t [belowduring(γ, t, D1, a(V), M1) ⇒ aboveduring(γ, t+E, D2, b(V), M2)]
ishigherduring(γ1, γ2, t, D, a(V) ≡
∀t1, V1, V2 [t≤t1≤t+D & at(γ1, t, a(V1)) & at(γ2, t, a(V2)) ⇒V1 ≥ V2]
higherleadstohigher(γ1, γ2, D1, a(V), E, D2, b(V)) ≡
∀t [ishigherduring(γ1, γ2, t, D1, a(V)) ⇒ ishigherduring(γ1, γ2, t+E, D2, b(V))]
higherleadstolower(γ1, γ2, D1, a(V), E, D2, b(V)) ≡
∀t [ishigherduring(γ1, γ2, t, D1, a(V)) ⇒ ishigherduring(γ2, γ1, t+E, D2, b(V))]

Based on these properties are formalized as follows.

HTDtoHBwithin(γ, D1, M1, E, D4, M4)

If in a trace γ for some time duration D1 the task demand is higher than M1, then after some delay E for some time duration D4 biases are higher than M4.

aboveleadstoabove(γ, D1, taskdemand(V), M1, E, D4, biaslevel(V), M4)

HTDtoHBbetween(γ1, γ2, D1, E, D4)

If in trace γ1 for some time duration D1 the task demand in γ1 is higher than the task demand in γ2, then after some time delay E, for some time duration D4 biases in trace γ1 are higher than biases in trace γ2.

higherleadstohigher(γ1, γ2, D1, taskdemand(V), E, D4, biaslevel(V))

HTDtoHXwithin(γ, D1, M1, E1, D2, M2)

If in a trace γ for some time duration D1 the task demand is higher than M1, then after some delay E1 for some time duration D2 the exhaustion level is higher than M2.

aboveleadstoabove(γ, D1, taskdemand(V), M1, E1, D2, exhaustionlevel(V), M2)

HTDtoHXbetween(γ1, γ2, D1, E1, D2)

If in trace γ1 for some time duration D1 the task demand in γ1 is higher than the task demand in γ2, then after some time delay E1, for some time duration D2 the exhaustion level in trace γ1 is higher than the exhaustion level in trace γ2.

higherleadstohigher(γ1, γ2, D1, taskdemand(V), E1, D2, exhaustionlevel(V))

HXtoLDCwithin(γ, D2, M2, E2, D3, M3)

If in a trace γ for some time duration D2 the exhaustion level is higher than M2, then after some delay E2 for some time duration D3 the demand of selected components is lower than M3.

aboveleadstobelow(γ,D2, exhaustionlevel(V), M2, E2, D3, componentdemand(V), M3)

HTDtoHXbetween(γ1, γ2, D2, E2, D3)

If in trace γ1 for some time duration D2 the exhaustion level in γ1 is higher than the exhaustion level in γ2, then after some time delay E2, for some time duration D3 the demand of selected components in γ1 trace is lower than the demand of selected components in trace γ2.

higherleadstolower(γ1, γ2, D2, exhaustionlevel(V), E2, D3, componentdemand(V))

LDCtoHBwithin(γ, D3, M3, E3, D4, M4)

If in a trace γ for some time duration D3 the demand of selected components is lower than M3, then after some delay E3 for some time duration D4 the biases are higher than M4.

belowleadstoabove(γ,D3, componentdemand(V), M3, E3, D4, biaslevel(V), M4)

LDCtoHBbetween(γ1, γ2, D3, E3, D4)

If in trace γ1 for some time duration D3 the demand of selected components in γ2 is lower than in γ1, then after some time delay E3, for some time duration D4 the biases in trace γ2 are higher than the biases in trace γ1.

higherleadstolower(γ1, γ2, D3, componentdemand(V), E3, D4, biaslevel(V))

Automated verification of these properties has been performed against generated simulation traces.

Dynamical System Models Used

Next, the overall executable cognitive agent model is addressed. It includes some computational models in dynamical system style (based on difference / differential equations) that are described here.

The cognitive model is based on literature from cognitive science and human factors research. Hancock and Meshkati (1988) define mental workload as: 'The operator's evaluation of the attentional load margin (between their motivated capacity and the current task demands) while achieving adequate task performance in a mission-relevant context.' An elaboration on their figure illustrating this principle is shown in Fig. 2.

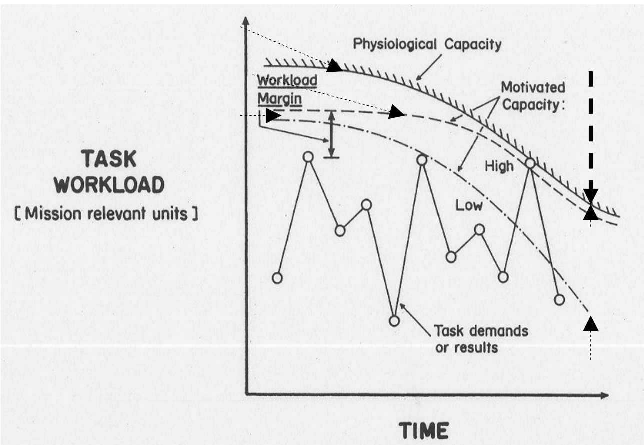

Figure 2: Cognitive processing levels over time

Basic concepts used and partly shown in this figure are:

$x(t)$	the exhaustion level at t
mp	maximal cognitive processing level if no exhaustion exists
rp	relaxed cognitive processing level if no exhaustion exist
$td(t)$	the externally determined task demand at t
$ptd(t)$	the internally perceived task demand at t
$em(t)$	the effort motivation level at t
β	parameter determining source of effort motivation
$ap(t)$	the available processing level at t
$cp(t)$	the current processing effort at t

The exhaustion level $x(t)$ is assumed to be normalized between 0 (no exhaustion) and 1 (complete exhaustion). As exhaustion affects possible processing levels, the maximal cognitive processing level at some time point t is taken to be $mp\ (1 - x(t))$, the relaxed cognitive processing level $rp\ (1 - x(t))$; this is also illustrated in Figure 2. The incoming external task demand td is transferred to the internal perceived task demand ptd by dividing it by the current maximal processing level ($mp\ (1 - x(t))$. When the result is above 1, it is set to 1 which ensures that the perceived task demand lies between 0 and 1. The perceived task demand and exhaustion level determine based on a personality characteristic parameter β what the current effort motivation level $em(t)$ is with a value between 0 (no motivation) and 1 (totally motivated). This level in return determines the current available processing level $ap(t)$, which influence the maximal processing effort $cp(t)$. See the next section for more details on these processes.

Exhaustion First the model for the level of cognitive exhaustion $x(t)$ over time is introduced. The exhaustion for a next time point depends on the current processing effort, but also on the current exhaustion, built up in the past. The assumption is that exhaustion increases proportionally with in how far the current cognitive processing effort $cp(t)$ is higher than the level indicated by $rp\ (1 - x(t))$. When the current processing effort is lower than this value, exhaustion decreases proportionally, until 0 is reached. Furthermore, the factor γ is used to fine-tune the model.

$$\Delta x = \gamma \frac{cp(t) - rp\ (1 - x(t))}{mp} \Delta t$$

$$x(t+\Delta t) = \begin{cases} x(t) + \Delta x & \text{if } x(t) + \Delta x > 0 \\ 0 & \text{else} \end{cases}$$

Effort Motivation At time t the cognitive effort chosen is assumed to be bounded by $mp\ (1 - x(t))$. Moreover, a personality characteristic parameter β is introduced indicating in how far the motivation for effort is externally driven through the perceived task demand (indicated by $\beta = 1$) or internally driven by the exhaustion level (indicated by $\beta = 0$). The effort motivation $em(t)$ is determined as follows.

$$em(t) = \beta\, td(t) + (1 - \beta)\ (1 - x(t))$$

Processing Level Made Available Given the motivation indicator the cognitive processing level $ap(t)$ made available is determined in the following manner. If the motivation is 1, the maximal possible processing level $mp\ (1 - x(t))$ will be the processing level made available, if the motivation is 0, the relaxed processing level $rp\ (1 - x(t))$. The general model for the processing level made available is:

$$ap(t) = (em(t)\ mp + (1 - em(t))\ rp)\ (1 - x(t))$$

When $cp(t) = ap(t)$ is taken (i.e., the processing level made available is fully used), the three models for $x(t)$, $em(t)$ and $ap(t)$ above can be combined to obtain a single (but complex) difference or differential equation model for $x(t)$, given the chosen values $cp(t)$ for the current processing effort over time.

Overall Cognitive Agent Model

This section describes the overall design of the cognitive agent model incorporating the dynamical models of the previous section. To evaluate whether the model indeed dynamically adjust its behavior based on external task demands in a way similar to humans, it has been designed in a formal, executable format. It includes various cognitive components and control knowledge about them. Moreover, it is able to observe the world, form goals, execute actions, and does this in the following order:

Determine Observations: The agent observes the world and forms beliefs about what it sees.

Determine Goals: Based on beliefs, it forms goals with priorities.
Determine Task Demand: Based on the formed goals, their priorities and the cognitive processing level that is required to reach them in the optimal way, the task demand is determined.
Determine Perceived Task Demand: The perceived task demand is determined from the real task demand (see section above).
Determine Effort Motivation Level: see section above.
Determine Available Processing Level: see section above.

After these processes the agent starts the selection process of the cognitive processing components to be executed.

Determine Executability of Components: First, it determines which components are eligible for execution, i.e. that they can actually produce outputs when selected. For this it checks for each component whether all the input it needs is available.

Determine Relevance of Components for Goals: Next, it determines which components are relevant for which goal:

```
∀g ∀c ∀k ∀cr ∀kr ∀x
If  goal(g) & component_has_output (c, g) &
component_requires_processing_level(c,k) & component_has_output (cr, g)
component_requires_processing_level(cr,kr) &
¬∃co ∃ko ( component_has_output (co, g) &
component_requires_processing_level(co,ko) &   ko > kr )
component_has_executability(c, 1) & exhaustion-level(x) &
b = 1 – | (1 – k / kr) – x |
then  component_has_relevance_for_goal(c, g, b)
```

This process entails that the relevance of a component *c* for a certain goal *g* that it has as its output depends on the current exhaustion level *x* and the existence of a most expensive component *cr* of all components with goal *g* as their output. The rational behind this process is that the most expensive component is the best (most rational) component to reach *g* and is preferred when there is no exhaustion (receives a relevancy of 1 - | (1 – kr/kr) – 0| = 1). However, the more exhausted the agent is, the more it prefers the cheaper components over the expensive ones. For example, when *x* is 0.3 *cr* only receives a relevancy of 0.7, while *c*, given it requires a lower processing level, e.g. 4 instead of 6, receives a relevancy of 1 - | (1 – 4/6) – 0.3| = 0.97. If a component does not have a certain goal as one of its outputs, its relevance for that goal is 0.

Determine Priority of Components: The priorities of the components for the various goals are determined by multiplying their relevancy for a goal with the priority of that respective goal.

Determine Components to be Activated: This is done by considering all possible groups of components for which it holds that 1) they have a priority greater than 0, 2) they are not relevant for the same goal, 3) their output does not make the goal of the other irrelevant. Furthermore, 4) their combined required processing level is below or equal to the available processing level. The components that are selected for execution are the members of the group with the highest total priority, which is formed by the sum of the priorities of the components.

Determine Activated Components: The components that are selected for activation are executed.

Determine Current Processing Effort: The current processing effort that the executing components deliver is determined.

Update Exhaustion Level: Given this current processing effort the exhaustion level is updated , see the section above.

As long as observations are made, the agent keeps controlling its process as indicated. When there is no task demand, the agent relaxes, resulting in decreased exhaustion.

Simulation Experiments

To evaluate the cognitive agent model simulation experiments were performed in a software environment especially developed for modeling executable temporal properties. For the evaluation a simple classification task was chosen because, although simplified, it is representative for the kinds of tasks future software agents might perform in, for example, training simulations of military air-traffic-controllers. The task entails the correct classification at every execution cycle of the objects (none, one or two) then present in the world. The classification of an object entails assigning it to one of the bins present in the environment. Objects have two properties, namely a color (red, blue or green), and a shape (cube, cylinder or triangle). This results in nine possible objects that can be classified. Bins also have two properties; it can either fit red, blue, green, or all colors and cube, cylinder, triangle, or all shapes. For the current scenario's it is assumed that these 4 bins are present:

Bin 1: fits red cubes
Bin 2: fits blue objects
Bin 3: fits any colored triangles
Bin 4: fits all objects

The general goal of the task is to classify objects, but also to do this as precise as possible. The assignment of an object to a bin whose properties it exactly matches has the highest preference. Furthermore, partial classifications are desired above an assignment to the most general bin. So in the current scenario the best classification of the red cube is assigning it to bin1, followed by an assignment in bin 4. A blue triangle can be assigned to bin 2 just as well as to bin 3; bin 4 however is less desired.

To test the behavior of the cognitive agent model over time, four scenarios have been developed. They all work with the same bins, but the objects present in the world over time differ. In the scenario named *1 object*, one object is present at every execution cycle. The similar principle holds for the scenario named *2 objects.* For scenarios named *low demand* and *high demand* the amount of objects varies, see Table 1 for an overview. Each scenario has length 9.

Table 1: Objects present in world over time

	1	2	3	4	5	6	7	8	9
Low demand									
High demand									

During the execution cycle of the model the agent first observes the world and forms beliefs about the properties of the available objects and bins, after which new goals are derived from the top level goal classify_all_objects as follows:

```
∀x ∀p
If   goal(classify_all_objects)
     goal_has_priority(classify_all_objects, p)
     belief(object, x)
then     goal(belief(classification_type_of,x, total))
     goal_has_priority(belief(classification_type_of,x, total), p/3 * 1.1)
     goal_satisfied_when(belief(classification_type_of,x,total),
                                                   belief(classified,x))
     goal(belief(classification_type_of,x, partly))
     goal_has_priority(belief(classification_type_of,x, partly), p/3)
     goal_satisfied_when(belief(classification_type_of,x, partly),
                                       belief(classified,x))
     goal(belief(classification_type_of,x, not))
     goal_has_priority(belief(classification_type_of,x, not), p/3 * 0.9)
     goal_satisfied_when(belief(classification_type_of,x, not,
                                                   belief(classified,x))
```

So for every object the agent forms three classification goals, with varying priority. These priorities express the agent's preferences for the various types of classifications.

The task demand for the current task is determined by the combined task demand of the present objects. Objects entail task demand because they cause goals with priorities.

$$\text{task\_demand} = \Sigma\{ mp * p \mid \text{goal}(g) \wedge \text{goal\_has\_priority}(g, p) \wedge \text{maximum\_required\_processing\_level\_for\_goal}(g, mp)\}$$

For the current task this entails that a single object delivers a total task demand of 5.06667. So for the scenarios *1 object* and *2 objects* the task demand will be constant; 5.06667 and 10.1333 respectively. For scenarios *low* and *high demand* the task demand varies, see Fig. 3.

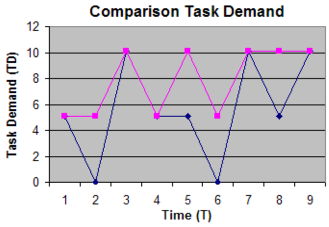

Figure 3: Task demand for scenario's Low Demand (blue diamonds) and High Demand (pink squares)

Above it was described how based on the goals and the priority of components for these goals, the cognitive agent model determines which components execute. Besides the components themselves it also uses control knowledge over these components, for example, their inputs and outputs and required processing level. The latter value is deduced from the number of required inputs of each component.

The following shows the process of a rational component in the form of an executable temporal rule:

If	then
HoldsAt (belief(object,x), t)	HoldsAt (belief(classified_as,x,y), t+1)
HoldsAt (belief(bin,y), t)	HoldsAt (belief(classified,x), t+1)
HoldsAt (belief(has_shape,x,s), t)	HoldsAt (belief(
HoldsAt (belief(fits_shape,y,s), t)	classification_type_of,x, total), t+1)
HoldsAt (belief(has_color,x,c),t)	
HoldsAt (belief(fits_color,y,c), t)	

This component requires a processing level of 6 and has a bias level of 0. Besides rational components, biased ones are present with a different process but a same output, e.g. :

If	then
HoldsAt (belief(object,x), t)	HoldsAt(belief(classified_as,x,y), t+1)
HoldsAt (belief(bin,y), t)	HoldsAt(belief(classified,x), t+1)
HoldsAt (belief(has_shape,x,s), t)	HoldsAt(belief(
HoldsAt (belief(fits_shape,y,s), t)	classification_type_of,x, total), t+1)

This component also deduces a belief about a total classification but forgets to take the color of the object and bin into account. The final result might be correct but might also be incorrect. This second component requires a processing level of 4 and has a bias level of 4/6, because the most expensive processing requires a level of 6; see the previous section on the relevance of components.

Last, various parameters present in the model are assigned a fixed value to arrive at an executable version. For the current task the maximal processing level *mp* is set to 10 and the relaxed processing level *rp* to 7. It is assumed that the agent is not exhausted at the beginning of the task. Furthermore, parameter γ, with which the granularity in exhaustion level over time can be tuned, is set to 0.3.

Each scenario was executed twice, once with personality value 0.7 (motivation primarily determined by external task demand) and once with value 0.3 (motivated primarily determined by internal exhaustion level).

Simulation Results

Scenario 1 Object In this scenario the cognitive agent model classified each object in a perfect way for both personalities. Since there is a maximum of one object at each execution, the maximal possible current processing level (for a total classification, i.e. the red cube) lies at 6. This is below the relaxed processing level, whose value is set at 7, and therefore no exhaustion occurs.

Scenario 2 Objects In this scenario the two objects ensure a constant high task demand of 10.1333. This results in a constant perceived task demand *ptd* of 1, which makes that both agents make more than their relaxed processing level available. Therefore the effort of the selected processing components often lies above the relaxed processing level *rp (1-x)*, causing the agent to become exhausted, which in turn influences the available processing level, see Fig. 4.

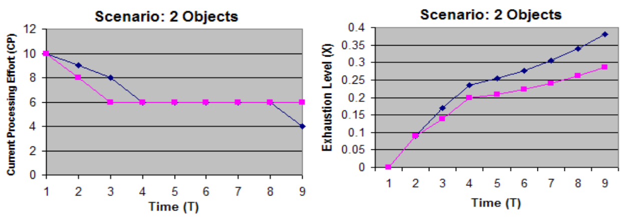

Figure 4: Current processing effort and exhaustion level for personality 0.7 (blue diamonds) and 0.3 (pink squares)

The agent with personality value 0.7 will, given the *ptd* of 1, make more processing level available than the agent with personality value 0.3. This is beneficial at first; more available processing level makes that more demanding, so less biased, components can execute. However, due to this higher effort level this agent becomes quicker exhausted, which results in that it over time actually has less processing level available, which result in the selection of cheaper and thus more biased components, see Fig. 5a.

Scenarios Low and High Demand In the scenarios *Low Demand* and *High Demand* the numbers of objects that are available at each execution cycle vary, see Table 1. The variety in the task demand, see Fig. 3, clearly determines the variety in current processing effort, see Fig. 5b. This in turn influences the exhaustion level and bias level, see Fig. 6.

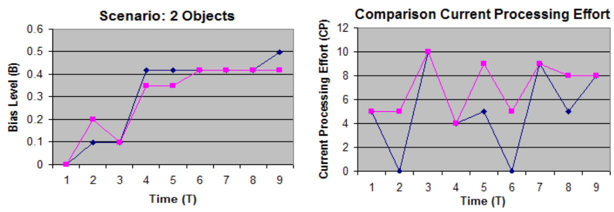

Figure 5: (a) Bias level for personality value 0.7 (blue diamonds) and 0.3 (pink squares) in the *2 Objects* scenario
(b) Current processing effort for the *Low* (blue diamonds) and *High Demand* (pink squares) scenarios

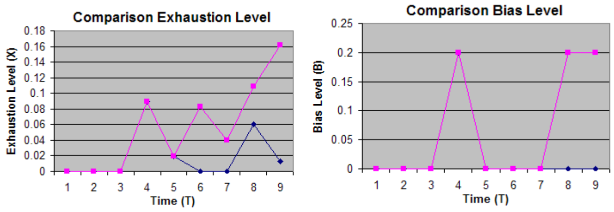

Figure 6: Exhaustion and bias level for the *Low* (blue diamonds) and *High Demand* (pink squares) scenarios

The increase in bias level has its impact on the quality of the task performance. Table 2 sums the percentage of false classifications of a single run averaged over all objects present for personality value 0.7. E.g., when the agent blindly assigned an object to any bin this is 75 percent, since it is correct for bin 4, which is one of four bins.

Table 2. Percentage of mistakes made by personality value 0.7

Scenario	1	2	3	4	5	6	7	8	9
1 object	0	0	0	0	0	0	0	0	0
2 objects	0	0	50	25	37.5	25	12.5	37.5	62.5
lowdemand	0	0	0	50	0	0	0	0	0
highdemand	0	0	0	50	0	0	0	100	75

Verification

Formalized properties such as those presented earlier have been automatically verified against a number of simulation traces, such as discussed above, using dedicated software developed for this purpose. As an example, property **HTDtoHBwithin** has been verified and shown to hold for all four traces for the following values for the duration and bound parameters: D1=100, E =100, D4=100, M1=8, M4=0.2. Notice that one execution cycle of the model takes a 100 time steps. Moreover the property **HTDtoHBbetween** which compares two traces was also verified and e.g., shown to hold for the low-demand high-demand scenario pair as well as the 1 object – 2 objects scenario pair for the values: D1= 100, E = 100, D4 = 100.

Discussion and Conclusion

This paper presented a cognitive agent model capable of dynamically adapting its behavior to external as well as its internal states. Related research with a similar goal focuses on integrating emotions, arousal, and motivation in cognitive systems, but no similar approach can be found. Closest to this work is the work of Ritter et al. (2007), who implements various theories of stress and their effect on behavior (some considered biases). However, the implemented factors were local, fixed and no temporal aspect is incorporated. One theory they did not implement is that tasks themselves are stressors (the approach taken in this paper). About this they state "we recognize that modeling tasks as stressors is an interesting and important next step in the effort to model the effects of stress."

The dynamical cognitive model was tested in simulations for various task scenarios. A formal analysis of properties of the model has been performed, including automated verification of the identified properties against simulation traces, indeed showing the behavior as expected.

For a number of choices that were made for the case currently presented also alternative choices could have been made, e.g., for the choice of parameters for the maximum en relaxed processing power in relation to the required processing level of components. It is expected that the values for the parameters depend on the application context. Based on the requirements on the behavior the model should show, these can be adapted as to provide a best fit.

As stated the model's goal is to be able to control the appearance of biases in a wide variety of tasks but even stronger, on more levels of the task execution. The current paper solely addressed the controlling of biases appearing in cognitive components processing beliefs. The processes from observations to beliefs and from beliefs to goals were fixed. These processes may just as well be subject to biases. A biased determination of priorities of goals can have serious effects on task execution. In future work the control of these processes will be added to the current model.

References

Baron, J. (2000). *Thinking and deciding (3d. edition).* New York, NY; Cambridge University Press.

Galton, A. (2006). Operators vs arguments: the ins and outs of reification. *Synthese, 150,* 415-441.

Gigerenzer, G., Todd, P. M., & the ABC Research Group (1999). *Simple heuristics that make us smart*. New York, NY: Oxford University Press.

Hancock, P.A., & Meshkati, N. (Eds.) (1988). *Human mental workload*. Amsterdam: North-Holland.

Hancock, P. A., & Warm, J. S. (1989). A dynamic model of stress and sustained attention, *Human Factors*, *31(5),* 519-537.

Johnston, W. A., & Heinz, S. P. (1978). Flexibility and capacity demands of attention. *Journal of Experimental Psychology: General, 107,* 420-435.

Kahneman, D. (1973). *Attention and effort*. Englewood Cliffs, NJ: Prentice-Hall.

Matthews, G., & Desmond, P.A. (2002) Task-induced fatigue states and simulated driving performance. *Quarterly Journal of Experimental Psychology, Section A, 55 (2),* 659-686.

McBride, S.A., Merullo, D.J., Johnson, R.F., Banderet, L.E., & Robinson, R.T. (2007) Performance during a 3-hour simulated sentry duty task under varied work rates and secondary task demands. *Military Psychology, 19 (2),* 103-117.

Posner, M. I., Boies, S. J. (1971). Components of attention. *Psychological Review, 78*, 391-408.

Ritter, F. E., Reifers, A. L., Klein, A. C., & Schoelles, M. J. (2006). Lessons from defining theories of stress. In W. Gray (Ed.) *Integrated Models of Cognitive Systems*. New York, NY: Oxford University Press.

Shields, S.A. (1983). Development of autonomic nervous system responsitivity in children: A review of the literature. *International Journal of Behavioral Development, 6*, 291-319.

Wickens, C., & Flach, J. (1988). Information processing. In E. Wiener & D. Nagel (Eds.) *Human Factors in Aviation*. San Diego, CA: Academic Press.