An Agent-Based System Assisting Humans in Complex Tasks by Analysis of a Human's State and Performance

Tibor Bosse[1], Fiemke Both[1], Rob Duell[2], Mark Hoogendoorn[1], Michel Klein[1], Rianne van Lambalgen[1], Andy van der Mee[2], Rogier Oorburg[2], Alexei Sharpanskykh[1], Jan Treur[1], and Michael de Vos[2]

[1]*Vrije Universiteit Amsterdam, Department of Artificial Intelligence*
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
{tbosse, fboth, mhoogen, mcaklein, rm.van.lambalgen, sharp, treur}@few.vu.nl

[2]*Force Vision Lab, Barbara Strozzilaan 362a, 1083 HN Amsterdam, The Netherlands*
{rob, andy, rogier, michael}@forcevisionlab.nl

Abstract

Human task performance varies depending on the task, environment, and states of the human over time. To ensure high effectiveness and efficiency in the execution of complex tasks, adaptive automated assistance of the human may be required. A personal assistant agent may constantly monitor the task execution and well-being of the human via non-intrusive sensors, and intervene when a problem is detected. In this paper, an agent-based system is presented that illustrates this approach. A human is given an increasingly complex task, while the future performance is predicted using observations and a dynamical model for the human's work pressure and exhaustion. If the predicted exhaustion becomes too high, the agent can assist the human in a number of ways. Experiments show that the support system increases performance with around 13%, and that it enhances the feeling of control of the situation.

1. Introduction

Human task performance can degrade over time when demanding tasks are being performed. Such degradation can for instance be caused by available resources being exceeded [16]. However, high effectiveness and efficiency levels are of particular importance for critical tasks. In such cases, automated assistance to support humans in effective and efficient task execution is often required.

The effectiveness and efficiency of the task execution depend on the capabilities, experience, and state of the person performing the task. Different persons as well as one person at different time points may require different degrees and types of assistance. To achieve this, an intelligent personal assistant is needed that monitors the actual performance of the user, takes his or her personal characteristics into account, and makes an analysis of the person's state for any given point in time.

In the past, a variety of intelligent personal assistants have been proposed to support humans during the execution of tasks (see e.g. [13, 14]). Such personal assistants usually include models that represent the state of the human and his or her tasks at particular time points, which can be utilized to determine when intervention is needed. An example of such a model addressing the cognitive load of the human can be found in [22]. Models differ in the considered aspects of human behaviour and of the execution of tasks. Depending on the model type, a specific personal assistant may react only to particular events and states of the human and his or her environment. Hence, a personal assistant that is able to choose and use different model types depending on the situation could provide a more proper form of assistance. To this aim, a generic agent-based architecture for an adaptable personal assistant has been proposed [3]. The architecture includes generic constructs that allow agents to perform self-configuration by loading domain specific models and thus altering their own functionality. The design of such a personal assistant includes a generic task model for monitoring, analysis and guidance of the human for which it can select the appropriate models given the goals that have been set, characteristics of the allocated tasks, and characteristics of the human.

This paper contributes an agent-based system that is an instantiation of this generic architecture. In addition, it is reported how the system was evaluated by a user study. The presented agent system functions as a personal assistant for a human that is given an increasingly complex task in the context of a (simplified) simulation-based training environment related to the military domain. While performing tasks in the training environment, non-intrusive measurements are made, e.g., the location of mouse clicks. Based on these observations and a model for work pressure [1], the personal assistant continuously predicts the person's state in the near future. If the predicted experienced pressure becomes too high, the person can be assisted in a number of ways, for example, by reallocating part of the task.

The remainder of the paper is organized as follows. Section 2 describes the military domain, which formed the motivation for the assistive system, and explains how the characteristics of a typical complex task in this domain were translated into the training environment. In Section 3, the general architecture and the specific design choices for the system are presented. Section 4 describes the setup of the user studies to evaluate the system, while in Section 5 the results of the evaluation are analysed. The paper concludes with a discussion.

2. Training in military domains

Although personal assistants can be beneficial in many areas, special interest goes to applications in which degradation of human task performance can have serious or harmful consequences for the human or the organization in which the human operates. Well-known examples of such applications in the public domain are driving assistance and personal health care. A challenging example in the military domain is naval combat management, which is the intended application area for the assistive system under development. Many combat management tasks performed by human operators are complex, because these tasks:

- must be performed within limited time,
- are critical for the organization,
- involve high volumes of information,
- take place in a dynamic environment,
- have a large solution space,
- require skill and expertise.

The combination of these characteristics can easily lead to demanding situations in which the human operator is likely to endure mental and physical aggravation, resulting in degradation of task performance. The aforementioned challenge lies in the fact that any personal assistant assisting these operators inherently is exposed to some of the same characteristics. For example, if the operator has limited time to perform some critical task, so does his personal assistant in assisting him.

Research on personal assistance for combat management systems is temporized by confidentiality of the military domain and limited access and availability of target systems. Hence, in early stages of research it is desirable for researchers to have an alternative setting that is similar to combat management systems, but doesn't have these disadvantages. Since combat management is considered an example of simulation-based training [21], a simulation-based training environment is the most obvious alternative setting for simulation and validation.

Based on the choice for a training environment, the next step is to translate combat management tasks into tasks that can be played in the simulation-based training environment. The approach taken here is as follows:

1. select a combat management task for which personal assistance is considered
2. classify this task in terms of task characteristics
3. invent a simulation-based training environment in which a task with similar characteristics is to be performed

The task characteristics used for classification of tasks are primarily based on workflow techniques for specification of tasks [4].

3. Architecture and design

This section describes the architecture of the personal assistant agent. An overview of this architecture is shown in Figure 1. Hereby, the box marked with agent represents the boundaries of the agent, in which several components are present. Furthermore, one additional element is present, namely the *external world*. The overall idea behind the various components and their interaction is described briefly below, after which each component is treated in more detail. The component within the agent which controls the overall process is the *coordination* component. This component provides control information to the other components. The component which is continuously activated is the *monitoring* component. This component gathers information about the human and the environment. These observations are passed on to the *analysis* component, which initially uses this information to determine the personal settings of the work pressure model, i.e. the values of the parameters in the model that are related to personal characteristics of human. The *analysis* component then determines what kind of observations are to be performed to verify whether the human is functioning as desired and transfers these as requests to the monitoring component. The observation results are transferred from the monitoring to the analysis component. This component makes an estimation of the work pressure which will be experienced by the human at some time point in the near future. In case this predicted work pressure is too high (i.e., beyond a certain boundary), the information about the prediction is passed on to the component *plan determination*. In *plan determination* actions are derived that can reduce the workload of the human such that the work pressure will not exceed the aforementioned boundary. Actions are for example task reallocation and giving task support. Below, each of these components will be discussed in more detail.

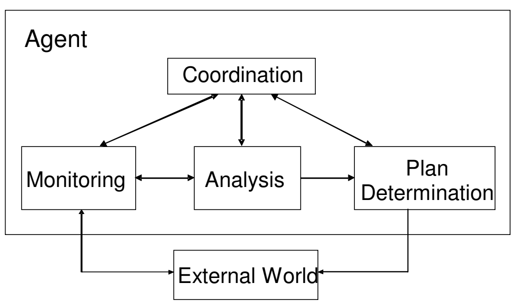

Figure 1. Agent Architecture.

3.1 Monitoring

The input and output ontology of the component consists of the predicates listed in Table 1. The monitoring component makes two types of measurements in the external world, initial one-time measurements and measurements during the simulation-based training program.

The one-time measurements determine the initial settings of the work pressure model for the expertise and personality profiles and basic cognitive abilities (further explained in Section 3.2). The first initial monitoring task is measuring four personality characteristics via questions from the NEO-PI-R and the NEO-FFI personality questionnaires [5]: extraversion, neuroticism, vulnerability and ambition. The second part of the initial measurements consists of three small tests. In the first test simple Reaction Time is measured (*choice*). In the second test (*calc*) humans have to solve mathematical equations. In the third test a human has to move the mouse to a target presented on the screen (*mouse*). In the second and the third test the human's speed and accuracy are measured. Data from all tests is transferred to the analysis component.

Table 1. Input/output ontology of monitoring.

Predicate	Explanation
Input	
to_be_observed: INFO_ELEMENT	The component is requested by *analysis* to observe a certain information element.
observation_result: INFO_ELEMENT x SIGN	An observation result is received from the *external world* indicating the information element observed, and the value thereof (i.e. pos or neg)
Output:	
to_be_observed: INFO_ELEMENT	The component outputs an active observation to be performed in the *external world*. Note that a translation occurs between the information requested by *analysis*, and the observations performed in the world.
observation_result: INFO_ELEMENT x SIGN	An observation result passed on to *analysis*. Again, a translation takes place in the monitoring component.

During the training program, the analysis component will request observations to be able to determine (using the work pressure model described in Section 3.2) whether the work pressure level of the human remains below the boundary. The observations required for this are the actions of the human in order to calculate performance quality and the situation on the screen in order to calculate situational task demand. These observations are requested by the analysis component continuously and will be sent back to analysis continuously.

3.2 Analysis

The analysis component is responsible for detecting problems in the desired functioning of the human. The input and output ontology is shown in Table 2. The component uses a dynamic model about work pressure to detect problems in the human functioning. The key concept in this model is the *experienced pressure*. The analysis component maintains a maximal desired value for this concept, called the *experienced pressure norm*. Using the work pressure model, the estimated experienced pressure for a time point in the near future (set to 3 minutes in the experiment) is continuously calculated. This calculation requires input about the functioning of the human. The analysis component therefore continuously requests information from the monitoring agent about the *task execution state*, which is measured via the scores that are received in simulation-based training environment. If the predicted experienced pressure will be higher than the norm, the analysis component will conclude that the experienced pressure norm will be exceeded and the following statement will be derived:

assessment(predicted_threshold_exceeded_by(exp_pressure, x))

Subsequently, this information will be forwarded to the plan determination component, which will plan the measures to be taken.

assessment(predicted_threshold_exceeded_by(exp_pressure, x))

Thereafter, this information will be forwarded to the plan determination component, which will plan the measures to be taken.

Table 2. Input/output ontology of analysis.

Predicate	**Explanation**
Input	
observation_result: INFO_ELEMENT x SIGN	An observation result is received from *monitoring*, indicating the information element observed, and the value thereof (i.e. pos or neg)
Output:	
to_be_observed: INFO_ELEMENT	*Analysis* outputs a request to observe a certain information element.
assessment: ASSESSMENT_VALUE	The component has assessed the current situation, and detected a problem. Therefore, the assessment is outputted, and passed on to *plan determination*.

The work pressure model (cf. [1]) is based on two different theories: 1) the cognitive energetic framework [8], which states that effort regulation is based on human resources and determines human performance in dynamic conditions; 2) the idea, that when performing sports, a person's generated power can continue on a *critical power* level without becoming more exhausted [7]. According to the model, a person's experienced pressure is influenced by a combination of *exhaustion*, the amount of *generated effort*, but also the *critical point*. The critical point is the amount of effort someone can generate without becoming more exhausted. In addition, the pressure is determined by external factors (task demands and environment state) and personal factors (cognitive abilities and personality profile) taken from the literature [17]. Due to space limitations, more details of the work pressure model have been left out. However, they can be found in [1].

3.3 Plan Determination

In case the *analysis* component has prospected that the experienced pressure will become too high within a certain period, the *plan determination* component derives actions to be performed such that the work pressure experienced by the human does not exceed the maximal experienced pressure which has been set.

Table 3. Input/output ontology of plan determination.

Predicate	**Explanation**
Input	
assessment: ASSESSMENT_VALUE	The assessment of the situation comes in from *analysis*.
Output:	
to_be_performed: ACTION	The *plan determination* component has derived that a certain action should be performed, e.g. reallocation of a percentage of the tasks.

The input and output ontology of the component is shown in Table 3. Below, two options are presented to determine how to intervene.

3.3.1 What-if simulation. In the first method called **what-if simulation**, *plan determination* uses the input from *analysis* concerning the predicted trend of the experienced pressure. Suppose the prediction is that the experienced pressure will become too high, namely $x1$ in y time units (which is above the boundary b that has been set). The component performs a what-if simulation using the work pressure model whereby the task level has been reduced by $z\%$ to investigate how effective such a change is. This results in a prediction that the experienced pressure will become $x2$ after y time units. Now, the *plan determination* component can calculate how much the task level should be reduced in order to stay precisely within the boundaries that have been set:

task_level_reduction_percentage = (x1 – b) / ((x1 – x2) / z)

The algorithm can be summarized as follows:

Algorithm: Action determination using what-if simulation

Input: y (the number of time units considered in the future); x1 (the prospected experienced pressure in y time points); d (the maximum experienced pressure); z (the standard percentage of task decrease to try)
Output: Task level reduction percentage

1. Given the prospected task level, calculate the value x2 of experienced pressure after time y if the level were to be reduced with z%.
2. return (x1 – b) / ((x1 – x2) / z);

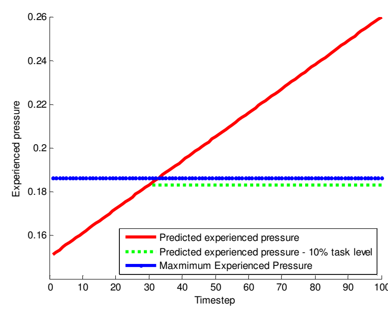

Figure 2. Plan determination what-if simulation method

Figure 2 shows an example of a prediction. The current time point is 30, and the boundary for maximum experienced pressure (b) is set to 0.186. Given the current trend in the task level, the experienced pressure is prospected to surpass the maximum level in 3 time units, whereby the predicted experienced pressure is $x1$ = 0.1863 to be precise. The *plan determination* component performs another what-if simulation whereby the task level is reduced with z = 10% in the future. This prediction is shown in Figure 2 as well. In this prediction, the experienced pressure with a task level reduction of 10% will be $x2 = 0.183$ after 3 time units. *Plan determination* has enough information to calculate how much the task level should be reduced at time point t, given this prediction: (0.1863 – 0.186) / ((0.1863 – 0.183) / 10) = 0.91%.

3.3.2. Fixed strategy. Another approach is to use a **fixed strategy** to reduce the task level. Hereby, a strategy is utilized which increases the reduction in task level with each consecutive time point at which an experienced pressure exceeding the maximum experienced pressure boundary *b* is predicted. The fixed strategy proposed in this paper is to decrease the task level with a percentage equal to the number of time units at which the prediction is that the boundary *b* will be crossed. As a result, an increasing decrease in task level will be performed until the prediction no longer forecasts exceeding *b*. So, at time point 1 the task pressure will be decreased with 1%, at time point 2 with 2% (in addition to the previous 1%), etcetera until the prediction is satisfactory.

```
Algorithm: Action determination using fixed intervention
Input: x (number of time steps that the prediction is above b)
Output: Task level reduction percentage
1. return x;
```

Given that it has now been determined that the task level needs to be reduced with a certain percentage, two possible approaches can be followed, namely *task reallocation*, and *task support*. In the former case, the reduction in task load is accomplished by reallocation of tasks. The tasks can be allocated to other humans, but the deadline of tasks could be switched as well (e.g. postponing tasks). The second option, *giving task support* reduces the task level by giving the human support, e.g. giving a calculator when having to perform calculations. Hereby, each support action reduces a certain amount of task level and hence, appropriate support can be chosen based upon the methods introduced above.

4. User Study

A user study was conducted to examine the effectiveness of the support given by the assistant. One experienced male participant took part, with the main task to perform tasks in a simulation-based training environment. The environment and the procedure of the study are described in Sections 4.1 and 4.2.

4.1 Simulation-based training environment

In the study, the main task consisted of identifying incoming contacts and, based on the outcome of the identification, deciding to eliminate the contact (by shooting) or allowing it to land (by not shooting). The object at the bottom of the screen represents the participant's (stationary) weapon. In addition, contacts (allies and enemies in the shape of a dot each accompanied by a simple mathematical equation) will appear at a random location on the top of the screen and fall down to random location at the bottom of the screen. The rate at which the contacts appear can vary in demanding versus less demanding circumstances.

The identification of a contact is performed by checking the correctness of its equation, incorrect equations correspond to enemies and correct ones to allies. Points are gained by shooting down the enemies and by allowing the allies to land. The participant can shoot a missile by executing a mouse click at a specific location; the missile will then move from the weapon to that location and explode exactly at the location of the mouse click. When a contact is within a radius of 50 pixels of the exploding missile, it is destroyed. The number of points a participant receives for destroying an enemy is proportional to the proximity of the explosion. When a participant shoots an ally or when an enemy reaches the bottom of the screen 10000 points are lost. When an ally reaches the bottom of the screen the participant receives 1000 points.

When support is given, the task-based reallocation, described in Section 3.3, is used. In this scenario, this indicates that when support is needed, a number of cases are reallocated (e.g., disappear from the screen). Support is given based on the (predicted) level of experienced pressure: if this exceeds 0.8 within 180 seconds the **fixed-strategy** (see Section 3.3) is used.

4.2 Procedure

The User Study consisted of two sessions; the first session was used to measure the participant's profile and the second session was used for examining the effect of the support system using the model.

For the first session, the participant started with the first part by filling out a personality questionnaire, which contained questions from the NEO-PI-R and the NEO-FFI [5]. Answers to these questions served as input for the participant's personality profile. After the questionnaire, the participant performed the three small tests in order to determine his basic cognitive abilities and expertise profile (as explained in Section 3.1).

The goal of the second session was to test the effectiveness of support using the personalized model. For this, the participant performed two conditions of the task explained in Section 4.1. In condition 1 support was given according to the personalized model.

In condition 2 no support was given. In both conditions the situational demands were high as every 2.25 to 4.5 seconds one contact entered the screen and the complexity of the equations was relatively high (e.g. 271/17=23). The participant started with condition 1 ("support") and after some rest he continued with condition 2 ("no support"). In both conditions the task duration was 10 minutes.

5. Analysis of Experimental Results

This section analyses the results obtained during the experiment. First, the key results are presented in a graphical manner. Thereafter a more formal verification of the experimental results is performed.

5.1 Comparison With and Without Support

In this subsection a thorough comparison is made between the two conditions of the experiment (i.e. with and without support).

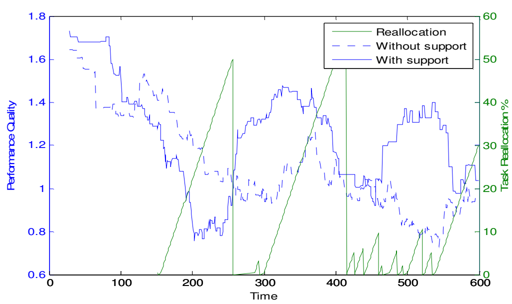

Figure 3. Performance quality with and without support

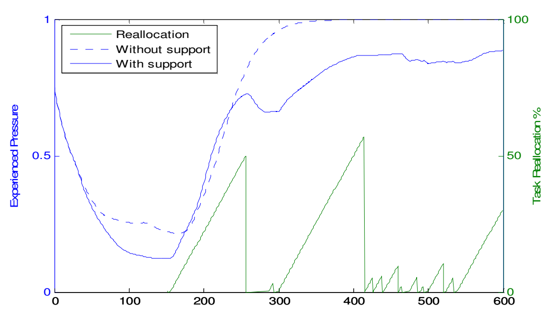

Figure 4. Experienced pressure with / without support

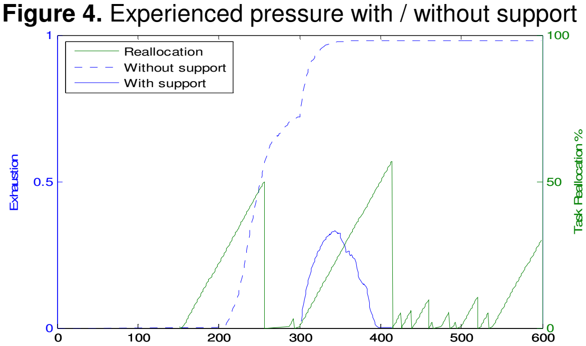

Figure 5. Exhaustion with and without support.

Obviously, one clear metric on the performance is the performance quality during the experiment. The results hereof are shown in Figure 3. On the x-axis of the figure time is shown in seconds, whereas the y-axis represents the performance quality (for the two dark lines, the scale is shown on the left side of the figure starting at 0.6), and the reallocation percentage (for the gray line, the scale is shown on the right side). Of course, the reallocation percentage belongs to the case with support, as no reallocation takes place in the other condition. It can be seen that the performance quality is generally higher for the case with support. Furthermore, as the performance quality decreases, the task allocation percentage increases, resulting in the performance quality going up again. Hence, the support is very effective. In addition the final score of the user in the support condition was much higher than in the no-support condition (600000 vs 50000 points).

Besides the performance quality, the internal concepts in the model also give an indication how good the support functions. Figure 4 shows the internal state experienced pressure for the case with and without support. It can clearly be seen that the experienced pressure is a lot lower for the case with support. The task reallocation stops the increase of the experienced pressure (and sometimes even makes it decrease).

Finally, in Figure 5 the internal state exhaustion is shown. Hereby, the exhaustion for the condition without support is far higher than the condition with support. Only between time point 300 and 400 there is some exhaustion in the case with support. In the other condition, exhaustion builds up, and never disappears.

5.2 Formal Analysis

Besides the analysis of the experimental results, the results have also been analyzed by verification of dynamic properties. Following [2], dynamic systems can be studied by specifying dynamic statements that are (or are not) expected to hold in terms of temporal logical expressions, and automatically verifying these statements against logs of the system. The purpose of this type of verification is to check whether the system behaves as it should. For the support system presented in this paper, a typical example of a property that may be checked is whether the performance of users is higher in the scenarios with support than in the scenarios without support. However, more functional properties of the system may be checked, such as 'every time that the system assesses that there is a risk for a critical situation, it will provide some form of support'. By verifying such properties, the developer can easily locate the sources of errors in the system.

For the support system presented in this paper, a number of such dynamic properties have been formalized in the language TTL [2]. This predicate logical language supports formal specification and analysis of dynamic properties, covering both qualitative and quantitative aspects. TTL is built on atoms referring to *states* of the world, *time points* and *traces*, i.e. trajectories of states over time. In addition, *dynamic properties* are temporal statements that can be formulated with respect to traces based on the state ontology Ont in the following manner. Given a trace γ over state ontology Ont, the state in γ at time point t is denoted by state(γ, t). These states can be related to state properties via the formally defined satisfaction relation denoted by the infix predicate |=, comparable to the Holds-predicate in the Situation Calculus: state(γ, t) |= p denotes that state property p holds in trace γ at time t. Based on these statements, dynamic properties can be formulated in a formal manner in a sorted first-order predicate logic, using quantifiers over time and traces and the usual first-order logical connectives such as $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\forall$, $\exists$. A special software environment has been developed for TTL, featuring both a Property Editor for building and editing TTL properties and a Checking Tool that enables formal verification of such properties against a set of (simulated or empirical) traces.

Based on this TTL language, various dynamic properties of the support system have been formalized. Below, a number of them are introduced, both in semi-formal and in informal notation:

P1 - Support Correctness

For all time points t,
if the system assesses that the experienced pressure norm will be exceeded
then within one time step the system will perform case reallocation.

P1(γ:TRACE) ≡
$\forall$t:TIME $\forall$x:REAL
state(γ, t) |= assessment(threshold_exceeded_by(exp_pressure, x))
$\Rightarrow$ $\exists$t2 [t ≤ t2 ≤ t+1 & state(γ, t2) |= to_be_performed(case_reallocation)]

This property has been checked against two traces γ. The first trace (which will be called trace1 from now on) was a log of the experiment under the "support" condition (see Section 4.2), whereas the second trace (trace2) was a log of the experiment under the "no support" condition. As could be expected, the property was satisfied for trace1, whereas it failed for trace2, which proves that the support system indeed performed an action in each case that it deemed this necessary.

As a next step, the effect of the support actions on the user's performance quality, experienced pressure, and exhaustion was verified. To this end, the average values of these concepts were compared for the different traces. The average value of performance quality, for example, can be calculated in TTL as follows:

P2 - Average Performance i

For trace γ, the average performance quality that a person has is i.

P2(γ:TRACE) ≡
[$\exists$i:REAL
i = $\Sigma_{t=o}^{TIME}$ $\Sigma_{p=o}^{REAL}$ case(state(γ, t) |=
has_value(performance_quality, p),p,0) / $\Sigma_{t=o}^{TIME}$]

As can be seen, this (logical) property requires counting of the different values of performance quality over time. For this, a very useful summation feature is available in TTL, denoted by $\Sigma_{k=0}^{t}$ case(φ(k), v1, v2). Here for any formula φ, the expression case(φ, v1, v2) indicates the value v1 if φ is true, and v2 otherwise. So, for the kth term, this summation adds v1 if φ(k) is true and v2 if φ(k) is not true.

In addition to P2, variants of the property have been specified in order to calculate the average values of experienced pressure and of exhaustion. The results of checking these properties to trace1 and trace2 are shown in Table 4. Note that the averages are calculated over traces with a length of 10 minutes, where each second a new value was logged. Thus, the averages are taken over 600 measurements.

As shown in Table 4, the agent's support increased the average performance quality (which is represented as a real number between 0 and 2, see also [1]) from 1.051 to 1.188, which is an increase of 13.0%. Moreover, the support decreased the (estimated) experienced pressure from 0.556 to 0.643 (13.5%), and decreased the (estimated) exhaustion almost completely: from 0.498 to 0.026 (94.8%).

Table 4. Average values of performance quality, exhaustion, and experienced pressure for 2 traces.

	Trace 1 (support condition)	**Trace 2 (no-support cond.)**
Performance quality	1.188	1.051
Experienced pressure	0.556	0.643
Exhaustion	0.026	0.498

6. Discussion

In this paper, an agent-based system was presented for highly personalized support during execution of a demanding task. The assistance provided by a personal assistant agent in this system is sensitive not only to the task and environmental conditions at hand but also to highly personal aspects such as the characteristics and states (such as work pressure and exhaustion) of the human at the given point in time. It constantly monitors the task execution and well-being of the human via non-intrusive sensors, and intervenes when an unsatisfactory situation is expected. Analysis takes place on the basis of continuously made predictions using observations and a dynamical model for the human's work pressure and exhaustion. Experiments have shown that the performance of a human increases up to around 13% in a scenario with support. In addition, the subject in the experiments reported that he had the feeling that he was able to handle the situation, in contrast to a scenario where no support was provided.

A number of approaches for prediction of human behavior in ambient intelligence environments have been developed. Most of these approaches perform time series analysis based on sensory data collected. In particular, data mining prediction techniques (e.g., case-based reasoning [11]), soft computing prediction techniques (e.g., fuzzy rule based learning [18]) and statistical modeling prediction techniques (e.g., based on Markov chains [9]) are used often. Usually cognitive models underlying human behavior are not considered in these approaches. However, combining information about observations of human behavior with dynamic specifications of internal processes, causing such a behavior, as proposed in this paper, provides a stronger basis for prediction.

To ensure correct prediction and appropriate support, cognitive specifications should be precise and valid. The work pressure model used in this paper has a strong support from psychology [15; 17; 19].

Some characteristics and behavioral modes of the human may vary during the task execution. To ensure high reliability of predictions and adequacy of the support provided by the assistant, real-time fine-tuning of the parameters of the work pressure model can be performed based on the observed human behavior and characteristics. To this end, real-time parameter estimation techniques can be applied, based on the global probabilistic optimization, gradient-based algorithms or filters [20]. In the future a dedicated component will be elaborated and added to the proposed agent architecture, which realizes one of these techniques.

7. References

[1] Bosse, T., Both, F., Lambalgen, R. van, and Treur, J. (2008), An Agent Model for a Human's Functional State and Performance. In: Jain, L. *et al.* (eds.), *Proceedings of IAT'08*. IEEE Computer Society Press, 2008, pp. 302-307.

[2] Bosse, T., Jonker, C.M., Meij, L. van der, Sharpanskykh, A., and Treur, J. (2009), Specification and Verification of Dynamics in Agent Models. *International J. of Cooperative Information Systems*, vol. 18, 2009, pp. 167-193.

[3] Bosse, T., Duell, R., Hoogendoorn, M., Klein, M.C.A., Lambalgen, R. van, Mee, A. van der, Oorburg, R., Sharpanskykh, A., Treur, J., and Vos, M. de (2009). A Multi-Agent System Architecture for Personal Support during Demanding Tasks. Technical Report, Vrije Universiteit Amsterdam, 2009.

[4] Both, F., Hoogendoorn, M., Mee, A. v.d., Vos, M. de, An Ambient Intelligent Agent with Awareness of Human Task Execution. In: Prodceedings of the 2008 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2008). IEEE Computer Society Press, 2008.

[5] Costa Jr, P.T. and McCrae, R.R., Revised NEO Personality Inventory (NEO-PI-R) and the NEO Five-Factor Inventory (NEO-FFI) professional manual, Psychological Assessment Resources, Odessa, FL (1992).

[6] Judge, T.A., & Ilies, R. (2002). Relations of Personality to Performance Motvation: A Meta Analytic Review. *Journal of Applied Psychology 87*, 797-807.

[7] Hill, D.W. (1993). The critical power concept. *Sports Medicine*, vol.16, pp. 237-254.

[8] Hockey, G.R.J. (1997). Compensatory control in the regulation of human performance under stress and high workload: a cognitive-energetical framework. *Biol. Psychology 45*, 73-93.

[9] Kaushik, A.R., B.G. Celler and E. Ambikairajah, A methodology to monitor the changing trends in health status of an elderly person by developing a Markov model, Engineering in Medicine and Biology 27th Annual Conference, pp.2171-2174, 2005.

[10] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P., Optimization by Simulated Annealing, Science, Vol 220, Number 4598, pages 671-680, 1983.

[11] Ma, T., and Y. Kim, Context-Aware Implementation based on CBR for smart home, IEEE Trans. Wireless And Mobile Computing, Networking And Communications, pp.112-115, 2005.

[12] Matthews, G. and Deary, I.J. (1998). Personality traits. Cambridge, UK: Cambridge University Press.

[13] Modi, P.J., Veloso, M., Smith S.F., and Oh, J., CMRadar: A Personal Assistant Agent for Calendar Management. In: Bresciani, P. et al. (eds.), Agent Oriented Information Systems II, LNCS 3508, Springer, 2005, pp. 169-181.

[14] Myers, K., Berry, P., Blythe, J., Conley, K., Gervasio, M., McGuinness, D.L., Morley, D., Pfeffer, A., Pollack, M., and Tambe, M., An Intelligent Personal Assistant for Task and Time Management. AI Magazine Summer 2007, pp. 47-61.

[15] Plomin, R., and Spinath, F.M. Genetics and general cognitive ability. Trends in Cognitive Science 6(4), 369-176 (2002).

[16] Posner, M. I., and Boies, S. J. 1971. Components of attention. Psychological Bulletin 78:391-408.

[17] Rose, C.L., Murphy, L.B., Byard, L., & Nikzad, K. The role of the Big Five personality factors in vigilance performance and workload. European Journal of Personality 16: 185-200 (2002).

[18] Rutishauser, U., J. Joller and R. Douglas, Control and learning of ambience by an intelligent building, IEEE Transaction on Systems, Man, and Cybernetics-Part A: Systems and Humans, vol.35, pp.121-132, 2005.

[19] Salgado, J. F. (1997). The five factor model of personality and job performance in the European community. Journal of Applied Psychology 82 (1): 30–43.

[20] Sorenson, H.W. Parameter estimation: principles and problems. Marcel Dekker, Inc., New York, 1980.

[21] Thurman, R.A. and Mattoon, J.S. (1994) Virtual reality: Towards fundamental improvements in simulation-based training. *Educational Technology* 1994, 56-64.

[22] Wilson, G.F., & Russell, C.A. (2007). Performance enhancement in an uninhabited air vehicle task using psychophysiologically determined adaptive aiding. Human Factors, 49(6), 1005-18.