Attention-driven Artificial Agents

Themis Balomenos(1), Nicolas Tsapatsoulis(2), Stefanos Kollias(2), Stathis Kasderidis(3) and John G. Taylor(3)

(1) R&D Dept., ALTEC S.A.
Fragoklisias 4, 15125 Maroussi, Greece
Phone: +30 210 6109746, Fax: +30 210 6109747
email: tmpal@mycosmos.gr

(2) Dept. of Electrical and Computer Engineering,
National Technical University of Athens
Iroon Polytechniou 9, 15773 Zografou, Greece
Phone: +30 210 7723037, Fax: +30 210 7722492
email: ntsap@image.ntua.gr, stefanos@cs.ntua.gr

(3) Department of Mathematics, King’s College London
Strand, WC2R2LS, UK
email: stathis@math.kcl.ac.uk

ABSTRACT: In many (if not all) of the domains that are related with machines' interaction with humans the human model is considered as the ideal prototype. Adaptation of the behaviour of an application as a function of its current environment known as context awareness is clearly one of these domains. The environment can be characterized as a physical location, an orientation or a user profile. A context-aware application can sense the environment and interpret the events that occur within. In this paper we present an attention-based model, inspired from the human brain, for constructing artificial agents. In this model adaptation is achieved through focusing to irregular patterns, so as to identify possible context switches, and adapting the behaviour goals accordingly. Simulation results, obtained using a health-monitoring scenario, are presented showing the efficiency of the proposed model.

KEYWORDS: attention control, context awareness, artificial agents, human brain

INTRODUCTION

In the era of Pervasive Computing artificial agents, hidden in information appliances, will be continuously running, in an invisible manner, aiming at the best fulfillment of human users needs. In this framework, artificial agents should be characterized by interaction transparency and context-awareness. Interaction transparency means that the human users are not aware that there is a computing module embedded in a tool or device that they are using. It contrasts with the actual transparency of current interactions with computers: both traditional input-output devices such as mice and keyboards and manipulations such as launching browsers and entering authentication information (by using a login and a password) are purely computer oriented. Context awareness refers to adaptation of the behavior of an agent as a function of its current environment. This environment can be characterized as a physical location, an orientation or a user profile. A context-aware agent can sense the environment and interpret the events that occur within it. Sensing the environment is very important for adapting the provided to the user services.

Attention is a very important attribute possessed by many animals. It becomes increasingly under voluntary control and less reflexive as the evolutionary tree is ascended. In this paper we consider the application of this facility to artificial agents. We consider how attention can be introduced into such agents, specifically those involved in the guidance of humans in tasks involving ‘wearables’, for context switch detection.

J. G. Taylor introduced attention in engineering control terms; this uses both inverse and forward models in order to optimise information processing used in decision-making. Here we implement an attention control architecture which shows how sensory control can be used for sensor based context capturing. A simplified health-monitoring scenario is used in simulations to show the response of the attention-driven agent, in particular by adaptively changing the sensor resolution, as well as taking into account the profile of the user.
THE USER MONITORING PROBLEM

In many circumstances artificial agents would have to monitor their users and decide on behalf of them for their welfare. To facilitate intelligent decision-making the artefacts should be able to adapt their monitoring strategy according to the context of the user. Due to (mainly) power and communication limitations sensors could not be continually polled in their highest sampling rate. This leads to the problem of limited information for effective decision-making. The controlling agents then have to adaptively monitor their users by increasing or decreasing the sampling rates of the various sensors involved. This is the User Monitoring Problem. It is an optimization problem in the sense that one has to balance energy and communication overhead against higher resolution information about the user and his environment. A possible strategy for achieving such a balance is attention control.

The principal idea behind such an approach is that user context switches correlate with higher probability with important changes in the user’s state. A context switch is a concept that indicates a transition from a previous equilibrium state to a new equilibrium. This concept is specialized further as it is applied in specific domains. However, this transition is the core property of detecting context switches. There are two principle ways in which such a state transition could take place. It is either a slow, gradual process (‘adiabatic’) or a ‘sudden’ jump to the new state. So, the rate of transition is the actual criterion for classification of the proper approach. Evolution has given to the various species the mechanism of attention. This is a process for detecting the ‘sudden’ changes. Typically a fast change indicates an increased uncertainty and as such is a prudent strategy to deal with it first. On the other hand slow changes are more easily captured by observation in the macro-scale, i.e. by considering general statistical characteristics of an appropriate population.

According to the above, we propose a solution to the User Monitoring problem, which consists of two elements. On one hand one uses attention control to capture the fast changes while classifier systems could capture the departure from one context to the other in the slow timescale.

Let us present now a simple health-monitoring scenario that would make more concrete the User Monitoring Problem. Here we have a user, which belongs to a special population group that of chronic patients, that needs to regularly monitor his health condition. There are three sensors attached to the user monitoring Heart Rate, Blood Pressure and Chest Volume. An artificial agent, residing in user’s PDA, polls the sensors, controls their sampling rate and informs the user. The PDA acts as the user interface and can also have the ability to call the health monitoring service provider in case of an emergency (serving as actuator).

A number of events can take place, which induce a change in the state of the user. We use the variable of Alert Level to distinguish the user states. The resting state is labeled Normal. Additionally two more states are considered: An Attention Seeking and a Dangerous one. The goal of the agent is to detect successfully the states and take appropriate actions.

If an event of interest takes place (e.g., Heart Attack or another major stress) we assume that the following cycle of events takes place:

1. Rest state 1 - Initial Steady State.
2. Pre-cursor state - indicating the onset of the event
3. Onset of event - say with duration of 1 min.
4. Rest state 2 - Final Steady State.

The first Rest State is the current equilibrium state where the system uses a default monitoring level for the user. All sensors are sampled on the default sampling rate. The second state is a pre-cursor signal which in many phenomena is present and for this reason useful to be exploited for expectation of system change. We make the assumption here that indeed this signal exists in order to simplify the presentation. The existence of a pre-cursor could be exploited by using appropriate rules for the sampling rate setting of the sensors. In a typical setting the sampling rate would be increased on expectation of the main event. In our case the sampling rate of the sensor that identifies the pre-cursor event is doubled (although doubling the sampling rates of all sensors could be also adopted). The onset of the event is manifested by the increase in the mean level of measures in (at least) one sensor. At the end of the event the rate returns to another resting state. During the main event the sampling rate for the sensor that captures the event is doubled again.

THE ATTENTION-DRIVEN ARCHITECTURE OF THE AGENT

The proposed attention architecture for an agent is shown in Figure 1. It is inspired from the human brain, where attention refers to two forms of control: sensory and motor response[9]. In the following we will look into the former that is going down to lower level data (down to the input sensors, so as to change sampling rate, for example) and sensory feedback in response to an attention signal. Details about each module are presented in the Implementation subsection, after a more general analysis of the overall architecture is given.
ANALYSIS

The sensor module is employed to capture data from biosignals generated by the user. In the current scenario, these signals include time series of the user’s Heart Rate (HR), Respiration Rate (RSP), Chest Volume (CHV), Systolic and Diastolic Blood Pressure (PS, PD). The Intermediate State Creator collects the signals, digitises them using appropriate sampling rates. Sampling rates are increased or decreased whenever a corresponding command is received from the Action Rules module. Obviously, if one increases the sampling rate, she/he will observe better the microstructure of the series. At this point, certain features that have diagnostic value in medicine are extracted and form the Intermediate State Vector, which is forwarded to the MAP module. At the same time, the sampled signals, referred to as Native Representation, are forwarded to the Observer module. This module includes a model for predicting the current state and future Intermediate States given the history Native Representations up to some lag time $T$.

At the next step, the predicted Intermediate State coming from the Observer, together with the actual Intermediate State coming from the Intermediate State creator, are fed to the Monitor module. There they are compared and if their difference exceeds some thresholds, an Attention Event is created for the responsible sensor. Thresholds are obtained from the Goals module. The main function of this module in the overall system is to indicate to the system the type of goals we try to achieve as well as to help in the creation and processing of new goals. In our case, however, its use is restricted to providing the abovementioned thresholds, which are user and context specific values. Attention Events are represented by corresponding Attention Indices (the strength of which is analogous to the inconsistency between the real Intermediate State and the predicted one).

In the MAP the Intermediate State Vector is transformed to the World State Vector; in our scenario the World State Vector corresponds to Alert Level classifications of user’s health state. Therefore, the MAP is a classifier system. In our implementation, the MAP corresponds to a hybrid intelligence system combining neural and neurofuzzy networks; the former provides the means for learning from numerical data and can be adapted to the peculiarities of a particular user while the latter provides the means for including a priori knowledge, in the form of rules, into the MAP.

The Attention Index together with the World State Vector is then forwarded to the Attention Controller. This module identifies the sensor to which attention should be given and decides which job will be dispatched in the next processing stage. Furthermore, it identifies mismatches between the results obtained from the MAP and the Monitor, and creates a reinforcement signal to the Observer. Finally, the Action Rules module defines the actions that should be undertaken in order to achieve the required Goals, thus, in our case, sends commands to the Intermediate State Creator regarding the adjustment of the sampling rate.

Figure 1: The proposed attention-driven architecture for an agent
IMPLEMENTATION

The architecture of Figure 1 has been implemented in SIMULINK to address the health-monitoring problem described in Section 2 (the code and installation guidelines can be found at [http://www.image.ntua.gr/oresteia/present/agent.zip](http://www.image.ntua.gr/oresteia/present/agent.zip)). Real sensor signals are emulated in the Sensors module. This module creates sensor values based on time series mechanisms. Adjustable parameters are the mean value of each signal, its standard deviation as well as the power of additive noise that may affect the sensor. Interrelations among signals are taken into account: Systolic and diastolic blood pressures are dependent as well as heart rate and respiration rate. Moreover, respiration rate is actually derived from chest volume signal.

The Intermediate State Creator consists of two modules: the Sampling and the Feature Extraction module. Increase or decrease in the sampling rate is performed whenever a corresponding command by the Action Rules module is received. Feature extraction aims at the derivation of measures that have diagnostic value in medicine. Currently the majority of features that are used are statistical ones obtained by using sliding time windows. Most of them correspond to features that medical experts use in their everyday practice. Output variables include:

(a) Three state vectors \(HR_SV, RSP_SV, Press_SV\). The following features are included in the state vectors:

\[HR_SV = \{HRinValue, HRmax-HRmin, HRmean, HRStd\}, \]
\[RSP_SV = \{RSPinValue, RSPmax-RSPmin, RSPmean, RSPStd, CHVmean, CHVStd\}, \]
\[Press_SV = \{PSinValue, PSmean, PStd, PDinValue, Pmean, PStd\}.\]

(b) Native Representation which is forwarded to the Observer.

The Observer module uses prediction models so as to forecast future Intermediate States. In its current implementation it predicts one sample ahead (the next Intermediate State) by using simple moving averages. The Goals module indicates the goals pursued by the Attention Controller in its current process. Currently, it just provides threshold values about the allowable deviation between real and predicted Intermediate State. These may depend on the particular user or context. All threshold values are adjustable.

The Monitor module identifies irregular patterns in the input space and produces Attention Indices for all sensors. It computes the Attention Index for each one of the sensors using the following relations (in the example below we consider only the blood pressure sensor). The feature vectors for the actual, the predicted, and the goal values for the blood pressure are:

\[Press_SV = \{PSinValue, PSmean, PStd, PDinValue, Pmean, PStd\} \]
\[Press_SVpred = \{PSinValuePred, PSmeanPred, PStdPred, PDinValuePred, PmeanPred, PStdPred\} \]
\[Press_SVgoal = \{PSinValueGoal, PSmeanGoal, PStdGoal, PDinValueGoal, PmeanGoal, PStdGoal\} \]

The Attention Index for this sensor is given by:

\[Press_AI = \max \{PSinValueAI, PSmeanAI, PStdAI, PDinValueAI, PmeanAI, PStdAI\} \]

\[0 \leq Press_ValueDev = |Press_Value - Press_ValuePred| = \frac{Press_Value - Press_ValueGoal}{\text{threshold}} \]

The Attention Controller is the most critical component of the architecture. It serves a variety of purposes: (a) Identifies the sensor to which attention should be paid, (b) It has a dispatch policy for deciding which jobs will be dispatched in the next processing stage, (c) Identifies mismatches between the results obtained from the MAP and the Monitor, (d) Creates a reinforcement signal, to be used by the Observer, in cases where the mismatch between MAP and Monitor is due to Observer’s malfunction. In more detail, in cases where anyone of the Attention Indices is greater than a given value (typically zero) it computes the highest and identifies the corresponding sensor. It forwards the current and the
previous value of the Attention Index of this sensor to the Action Rules module. At the same time stores the value of the Attention Index, for any other sensor that requires attention, in the priority stack. If all attention indices are null then forwards the current and the previous value of the Attention Index of the sensor found first (if any) in the priority stack. It checks the consistency between the World State and the Attention Index. For example if the World State shows Normal state while the Attention Index for a particular sensor is high for a long time this indicates a possible problem either to the MAP or to the Observer (bad prediction). If the inconsistency between MAP and Monitor is due to fault operation of the Observer a reinforcement signal is created. In its current form the Attention Controller is not able to identify which module to blame (Observer or MAP).

The Action Rules module defines the actions that should be undertaken in order to achieve the required Goals. In the case of a mismatch between the current and previous value of the Attention Index it creates an Alert Level which causes commands for increasing or decreasing the sampling rate of the attended sensor to be activated. Although in cases where the current value of the Attention Index is significantly higher than the previous one the command is indeed to increase the sampling rate, the opposite is not that simple. Decreasing the sampling rate might be ordered only if power consumption needs to be reduced.

The MAP module maps the Intermediate State to the World State. MAP is currently implemented using the CAM/SPM model presented in [10]. The CAM module partitions the Intermediate State space so as to produce linguistic terms (like HR_high). Linguistic terms are then used in the SPM module to activate the rules that have been modelled. Only rules corresponding to Dangerous and Attention Seeking states are included in SPM. This conforms to the medicine practice where healthy means not ill. Currently 16 rules have been modelled; nine correspond to Attention Seeking and seven to Dangerous.

SIMULATION AND RESULTS

In this section we present results from the SIMULINK implementation of the attention-driven architecture for agents shown in Figure 1. The following assumptions have been made:

1. A User Profile exists for guiding the system, i.e. providing the thresholds for the specific user.
2. Three time series are used, those of the Heart Rate, Blood Pressure and Chest Volume. Features extracted from all are fused in the MAP module, where rules are activated in order to reach useful decisions.
3. A State Classifier (MAP) has been implemented using the CAM/SPM model [10] that distinguishes the classes of events described above.
4. All signals that are used are synthetic.

![Heart Rate prediction vs. real](image)

**Figure 2:** The sampling rate changes as a response to the difference between the predicted and the actual values of the heart rate
During the simulation we are able to provide the system with data from a variety of cases, from normal to simulated heart attacks so as to check and evaluate its operation. We have found that in the majority of the cases the system responds rapidly and reaches accurate decisions regarding the user’s health status.

Figure 2 illustrates the context switch in the case where the predicted value of the signal is significantly different from the actual value. It should be noted that there are two possibilities for such deviations to occur. On one hand, either the model is correct in its prediction and indeed the actual state is different from what was expected, or the model prediction is simply wrong and nothing out of the ordinary has happened. The first case is distinguished from the second by using the classification history trace of the State Classifier. If the history indicates that we do have a change in the state classification we probably deal with the former case. Otherwise the Observer model has failed and needs further estimation. This incremental improvement can be implemented as either a supervised learning scheme (off-line) or as a reinforcement scheme (on-line) using the Monitor Error level. In the case of Figure 2 an event of interest has actually happened and therefore when the actual value of the heart rate (blue line at the upper plot of Figure 2) starts to divert from the one predicted by the Observer (red line), the sampling rate is increased to correspond to the warning state (lower plot of Figure 2). It can be observed that the response of the system is quite fast. As the difference is further increased, the sampling rate is accordingly increased to the attention state. The increase in the sampling rate is essential for capturing crucial details in the heart rate time series in order to reach more accurate decisions regarding the user’s health status. As the heart rate drops and approaches the predicted values, the sampling rate decreases accordingly.

![Systolic Blood Pressure](image1)

![Attention Index vs. time](image2)

Figure 3: The Alert Level changes between Warning and Danger as the blood pressure time series evolves in time

Figure 3 is an example of how the system handles possibly dangerous situations. In this case, the user’s blood pressure increases significantly above normal. The system responds by activating the Attention Index for the blood pressure sensor as well as changing the Alert Level. The first results in increased attention to the blood pressure time series, i.e. sampling it at higher sampling rates, while the latter may lead to issuing warnings to the user or taking other actions in the case of an emergency. More specifically, in the upper plot of Figure 3 we can see the time series of the Systolic Blood Pressure. As soon as the blood pressure starts to increase, the MAP module fires a Warning as it can be seen at
the middle plot of Figure 3 (blue line). When the blood pressure exceeds 150 mbars the Alert Level changes to Danger (red line). As blood pressure drops to 150 mbars again the Alert Level changes again to Warning. It can be observed that the system reaches timely and accurate decisions regarding the user’s health status. The Attention Index for the blood pressure sensor shown at the lower plot of Figure 3 instantly captures the potentially dangerous increase in the blood pressure values and returns smoothly to normal level (zero) as the blood pressure drops back to normal. This ensures that attention continues to be paid to the blood pressure sensor for some time so as to keep on capturing details in the time series in case the blood pressure increases again above normal.

Figure 4 illustrates the response of the system in the case of inconsistency between the MAP module and the Attention Index. The fact that the values of the Attention Index remain at relatively high values (lower plot of Figure 4) although the danger has passed according to the MAP module (middle plot of Figure 4) results in the reinforcement signal of the upper plot in Figure 4. This could be an indication that either the MAP module or the Observer needs retraining. At this point we haven’t developed a way of distinguishing which module should be retrained.

**CONCLUSIONS AND DISCUSSION**

The attention-driven architecture for an agent that is presented in this paper is part of the ORESTEIA Architecture for attention-based agents. There, an agent is composed by a number of artefacts. The artefacts are partitioned in four levels. At the first level the sensors and actuators exist. In the second one pre-processing facilities are provided. The third level involves local decision makers and the fourth one has global decision makers. The architecture presented here is for the most part the architecture of the level 3 ORESTEIA artefacts.

The system’s response to the changes in the time series of the user’s biosignals is very fast and the decisions regarding the user’s health status are accurate. Improvements in the Goals module so that it indicates to the system a range of goals to be achieved instead of just providing thresholds, a better implementation of the Observer, as well as a
mechanism for distinguishing whether it is the MAP module or the Observer the one that needs retraining in the case of inconsistencies are currently under development. Additionally, the collection and processing or real physiological data is being performed in order to provide the system with real data instead of simulated ones and test its behaviour in that case.

Finally, learning has to be present to enable the system to adapt better to its environment. Adaptation can be achieved by improving the classification power of the MAP module and the prediction ability of the Observer. The latter can be used to predict further into the future so that the system demonstrates a more proactive behaviour than it currently has.

Acknowledgment: The work presented in this paper has been undertaken in the framework of the ORESTEIA project (Modular Hybrid Artefacts with Adaptive Functionality, IST-2000-26091) [11].

REFERENCES