A Formal Model for Embedded Brain Reading

Elsa Andrea Kirchner
Robotics Lab, University of Bremen
and
Robotics Innovation Center, German Research Center for Artificial Intelligence (DFKI)
28359 Bremen, Germany
ekir@informatik.uni-bremen.de

Rolf Drechsler
Group of Computer Architecture, University of Bremen
and
Cyber Physical Systems, German Research Center for Artificial Intelligence (DFKI)
28359 Bremen, Germany
drechsler@uni-bremen.de

Abstract

The presented work contributes to research in the field of advanced man-machine interaction and to research in the field of formalisation and verification of complex systems. The paper describes for the first time *embedded brain reading (eBR)* and presents a formal model for it. It illustrates how an error-prone approach like brain reading (BR) can be applied safely by embedding it into the control of a real system and by applying mechanisms that control for its correct function. This work was motivated by the need to provide a detailed and well understandable formal description of eBR. We first introduce eBR and point out its main features. Next a general model for eBR is developed to describe the overall architecture, integral parts and dependencies between those parts. The model is developed and presented in a formal structured form that allows for application of optimisation as well as verification techniques. We demonstrate using implementations that the application of the formal model allows to check for completeness and correctness to detect errors in implementations, which were invisible without formalising eBR. In summary, the presented work contributes a formal model for a complex system and shows that such a formal model can improve the overall system's functionality. For future work our results support the application of formal modelling and verification techniques at the system level and the development of methods to prove for correctness and completeness of complex systems during their development.

keywords man/machine interface, rehabilitation, robots, telepresence, inspection and testing

1 Introduction

To interact intuitively with robotic systems in complex application scenarios advanced human-machine interfaces (HMIs) are required that support humans on demand, allow intuitive interaction, and provide intelligent operator support. This becomes more important as the complexity of scenarios increases. In Fig. 1 and 2 (multi-) robot telecontrol scenarios are depicted. In such complex control scenarios an operator has to simultaneously telecontrol one or a group of robots while he also has to respond to information that are relevant for the controlled systems and for the interaction with other people.

For advanced human-machine interaction, intelligent and adaptive interfaces must fulfil the changing requirements of an operator during interaction. Knowledge about the operators requirements and intentions can be gained by interpreting his brain state with respect to his current behaviour and situation. To enable this, brain activity of a human can be recorded by means of different methods, like the electroencephalogram (EEG) or functional magnet resonance imaging (fMRI). To analyse such data brain reading (BR) can be applied. BR was introduced as a method to gain information about hidden processes and states of the brain, i.e., the function of the mind (Coles, 1989). In Haynes and Rees (2006) it could even be shown that BR can be applied to detect different conscious states of the human, i.e., in his conscious perception. However, more functional questions like the decoding of visual, auditory, perceptual or cognitive patterns are addressed as well. Most studies are investigating fMRI data as for example in Kamitani and Tong (2005), in Miyawaki *et al.* (2008), in Naselaris *et al.* (2009) or in Polyn *et al.* (2005), although some work is also done on EEG data (Coles, 1989; Suppes *et al.*, 2009).

For our purpose, we want to define BR as the *online* decoding of brain activity into otherwise hidden information about the users functional or cognitive state, in this paper referred as brain state, with respect to internal or external events that are relevant for the human-machine interaction. This decoding takes place unnoticed by the user and requires no extra attention or cognitive resources of the user it is applied to. To apply BR for human-machine interaction, it is further required that BR enables the single trial, online analysis of brain activity of a human that is interacting freely with his environment. Our definition of BR does in principle also cover invasive methods for brain activity recording. However, invasive methods are not yet generally applicable in many fields of human-machine interaction and will thus not be considered here. Furthermore, in the given examples we focus on EEG data, since it is in comparison to other methods easy to record and to apply for the improvement of human-machine interaction.

Similar to BR the implementation of *passive* brain-computer interfaces (BCIs) that are for example applied for error detection in human-machine interaction (Zander and Kothe, 2011) requires the analysis of passively evoked brain activity and provides information about the humans intentions, situational interpretations or emotional states (Zander and Kothe, 2011). However, many passive BCIs are implemented to detect brain states that are related to behaviour which was performed *previously*, e.g., the detection of error related brain activity after a response error was performed.

The outcome of BR, i.e., detection of a current brain state, is for its application in human-machine interaction used to interfere on *future* behaviour of a human during interaction. This is important, since only by interfering *future* behaviour, HMIs can then be adapted with respect to the inferred behaviour to enable a more intuitive interaction with better support. Figure 2 depicts how BR can be applied to detect the brain state of ”movement preparation” for the adaptation of an interface, i.e., exoskeleton, to better support the *future* behaviour of ”movement onset” which is inferred from the output of BR [1]. Before movement onset (black vertical line in inserted diagram) continuous classification of the EEG by BR shows an increasing score for the likelihood of the brain state ”movement preparation” (blue curve in inserted diagram). Based on the score values the exoskeleton’s control is adapted to better support the inferred behaviour ”movement onset”.

However, the output of BR has to be taken with care since the classification of the brains activity by means of methods available today and by the fact that a human is a very complex system which is hard to model, cannot be done error free. Hence, control mechanisms have to be implemented that assure reliability of predictions or check for their correctness. We argue that predictions made based on the outcome of BR analysis should not be used to directly control a machine, but to adapt its control-interface. This adaptation should not be implemented such that a control command is for example reverted, as it is usually the case in error potential based BCIs (Ferrez and Millán, 2008; Seno *et al.*, 2010), since also the detection of errors from the EEG can be incorrect and can result in an erroneous correction of a command. In the given example, the detection of the brain state ”movement preparation” is hence not used to directly change the behaviour of the exoskeleton. Instead, it is applied to adapt its control. The more likely movement preparation is detected by BR, the shorter the user has to press against the force sensors that release the system from a locked in rest mode to a free run mode. In detail, the output of BR is used to change the time-threshold of the force sensors that have to be triggered for the release of the system from rest mode. This adaptation of the control allows a faster response of the exoskeleton on the inferred change in behaviour, i.e., start of

[1] see video ”Movement prediction for exoskeleton control” referenced in Section Supporting Media

movement, and reduces the force that has to be applied by the operator. However, the integrated force sensors have to be triggered to allow a change of mode, hence the inferred behaviour must *always* be confirmed by the sensors (Folgheraiter *et al.*, 2012). This prohibits faulty system behaviour in case of misclassification of the brain activity by BR or in case of a false inference about an upcoming behaviour.

From the above discussion it can be generalised that BR has to be fully integrated into the control of an HMI to enable its safe adaptation. For such *embedded BR* (eBR), an automated description of the current situation and behaviour of the human with respect to the interaction that is taking place is required. Thus, the behaviour of a human has to be analysed constantly and in an online manner to automatically mark or label situations and behaviour that are relevant for the interaction. For this labelling of data not only sensors that are integrated in the HMI but also additional physiological or psychophysiological measures (Coles, 1989) or information about the situation that is gained from the analysis by other supportive systems can be used. In the given example, instead of using sensors to confirm the inferred behaviour, the detection of muscle activity by the analysis of the electromyogram (EMG) (Hodges and Bui, 1996; Kirchner *et al.*, 2013; Tabie and Kirchner, 2013) could also be used to confirm movement onset. Further, a combination of different (psycho-)physiological measures, like EEG and EMG (Kirchner and Tabie, 2013) , can improve the quality of the predictions and thus the behaviour of the HMI to enable the HMI to adapt to different conditions.

The main challenges for improving man-machine interaction by eBR is the implementation of complex rules and requirements: complex application and communication rules, robotic control mechanisms, system design and rules for interpreting EEG data with respect to brain states and for the inference of upcoming behaviour as well as the application of control mechanism that assure a fault free behaviour of the whole system after the integration of eBR. To give a good description for such a complex system is challenging and especially difficult if scientists from different fields are involved. Errors in implementations are under such conditions hard to avoid. This motivated us to deduce a general, formalised model for eBR that can be applied to different implementations (see Section 2). We do not make use of a formal specification approach, but we introduce a rigorous specification of all interfaces. Out of this a formal description, as e.g., a Kripke structure, can directly be derived. Based on the investigations that were performed for this paper we describe in Section 3 that the developed and applied formal model improved our approach by (1) contributing a detailed description of the system, (2) optimising underlying procedures, (3) enhancing general reproducibility and (4) improving comparability with similar approaches, (5) pointing out small but relevant differences between approaches that cannot be derived otherwise, and by (6) easing the detection of errors in implementations. Our results are summarised in Section 4, where we discuss why the application of formal modelling and verification techniques at the system level (Drechsler and Große, 2005; Tabakov *et al.*, 2008) is important for different fields of application and for pursuing new paths in advanced human-machine interaction.

2 A Formal Model

In this section we present a general formal description of eBR for the adaptation or control of HMIs. Figure 3 depicts the developed formal model for eBR, which requires HMIs that not only control the application or robotic system but can be adapted by predictions made based on the analysis of brain activity to better support humans during interaction and can further control, i.e., drive and correct, eBR. Furthermore, the HMI itself or supportive systems must detect behaviour or situations that are relevant for the interaction in order to label the digitised EEG, i.e., to mark instances in the EEG data or situations during interaction that have to be analysed by eBR. The eBR system does analyse the relevant EEG data which is recorded at time $t = i$ to detect brain states that allow the inference of future behaviour at time $t = q$, with $i < q$. To improve eBR, other systems can be integrated that allow the analysis of other data (e.g., EMG and eye movement) to detect human behaviour that might be relevant for the interaction. These supportive systems can either produce markers for BR and hence trigger BR (bold lines from *MG* in Fig. 3) or give feedback to the HMI and BR system, to, e.g., approve the behaviour that was predicted (dotted lines pointing towards the HMI in Fig. 3) to control or to correct BR, i.e., to decide whether the output of BR is or was valid.

To apply eBR in a specific application, two rules have to be defined: (1) R^{BR} for processing of brain activity and prediction of the brain state and (2) R^{AHMI} for inferring on future behaviour (based on the predicted brain state) as well as for the adaptation of the HMI to better support the inferred behaviour and to control for correctness of the inference that was made. Both, the first and the second rule together define *how* BR is *embedded* into the system. However, the BR rule R^{BR} can in most cases be kept the same for different applications as long as the same brain states have to be detected and the same labels are provided by the HMI, supportive systems or BR system itself (Section 2.2), while the adaptation rule R^{AHMI} is likely different for different applications and depends on the way the output of BR is used to improve man-machine interaction. In the following we formalise all parts of the model as depicted in Fig. 3. Different parts of the model are relevant for BR as well as for embedding BR (REC, MG, WS, and SP in Fig. 3) and others are only relevant for eBR (PB, A, and C in Fig. 3).

2.1 Recording and Analog-Digital Conversion of Brain Activity (REC)

As explained before, any method that allows fast recording and analysis of brain activity can be used for BR. Here, we will focus on the EEG as source for brain activity for the given reasons. First, the analogue brain signal A ($A \in \mathbb{R}^c$), which is recorded with standard recording devices with c channels (electrodes) from the brain at time t ($t \in \mathbb{R}$), is transferred into a digital output signal

$$O(t) = \begin{bmatrix} d_1(t) \\ d_2(t) \\ . \\ . \\ d_c(t) \end{bmatrix}, \tag{1}$$

with $d_l \in \mathbb{N}', \mathbb{N}' \subset \mathbb{N}, l \in [1, c]$ and $\mathbb{N}' = \{-2^u, -2^u + 1, ..., 2^u\}$, with $u \in \mathbb{N}, u = z - 1$, with $z \in \mathbb{N}$, where z is the bit with and c is the number of channels, $t = n\Delta t, n \in \mathbb{Z}$ and Δt is the sampling interval.

The analogue-to-digital conversion (ADC in, Fig. 3) takes place on hardware side and is hence, dependent on the hardware that is used, e.g., bit width z of the AD-converter. The signal is sampled with a hardware specific sampling frequency $f = 1/\Delta t$. For a certain time point i the output of the AD-converter is

$$o(i) = \begin{bmatrix} d_1(i) \\ d_2(i) \\ . \\ . \\ d_c(i) \end{bmatrix}. \tag{2}$$

2.2 Labelling of Behaviour and Situations: Marker Generation (MG)

To analyse the digital signal by eBR to detect brain states certain segments or windows of the signal $O(t)$ have to be chosen for further processing. The choice is made based on the relevant current behaviour or relevant situations during interaction that allows and requires the prediction of future behaviour. In the example of the adaptation of the exoskeleton's control by eBR (Fig. 2), the change from rest mode to teleoperation is a relevant situation. During rest mode it is relevant to detect the brain state "movement preparation" by BR to infer on the behaviour "movement onset". For training of the classifier in BR this change of behaviour is automatically detected by sensors in the exoskeleton and labeled in the EEG data. For the application of eBR it is only relevant to automatically detect the onset of a rest situation to trigger BR for the detection of "movement preparation". For training and application, automated labelling of EEG data with respect to mode changes is done by the HMI.

As mentioned before, other supportive systems that analyse interaction behaviour based on other data, like the EMG in the example given in Fig. 2, which can also be analysed to detect movement onset, i.e., the change from a rest mode to teleoperation mode, can be used. In the example shown in Fig. 1 EOG or eye movement detection by eye tracking can be used to detect the focussing of important target objects like warning messages that require a response of the operator to trigger BR analysis for the detection of brain states. Depending on the detected brain state an assumption on whether these focused warnings have been perceived as important by an operator or not is possible. These assumption about the perceptional state of the operator then allows to infer on possible future behaviour, i.e., the likeliness of response behaviour of the operator on important target objects. Based on the inferred behaviour the HMI called "operator monitoring system" that is applied in the application depicted in Fig. 1 can than be adapted with respect to the tolerated response times (Kirchner *et al.*, submitted)[2].

Furthermore, markers can be generated based on the outcome of the analysis of recent EEG instances by BR to adapt an classifier (Wöhrle *et al.*, 2013). Important is that any marker is generated in an automated fashion based on predefined rules, e.g., in the first example the behaviour of movement onset is detected only in case the operator is in a rest position, since only in this situation movement onset is relevant. In all cases, to mark relevant behaviour or situations at a time point $t = i$ in the EEG stream, the digital signal $o(i)$ is hence labelled by an automatically generated marker $m(i)$ of a certain type. A marker $m(t)$ is specific to the application or type of interaction and can be defined as

$$m(t) \in M^{Apl}, M^{Apl} = \{-2^u, -2^u + 1...2^u\}, \tag{3}$$

with one type of $m(t)$ for *no* marker.

In some cases it is required to cut several windows with respect to a certain marker that was provided by the HMI or an additional supportive system to label a behaviour or situation. To enable this, additional markers might be added that define the distance between the windows. These markers are introduced by the eBR system itself as defined in the BR rule R^{BR}.

Adding markers to each output $o(i) \in O$ results in the signal $o_m(i)$ defined as

$$o_m(i) = \begin{bmatrix} d_1(i) \\ d_2(i) \\ . \\ . \\ d_c(i) \\ m(i) \end{bmatrix}, \tag{4}$$

with $m(i) \in \{-2^u, -2^u + 1...2^u\}$.

2.3 Windowing of Relevant Instances (WS)

For processing, the labelled data has to be segmented into windows that must have a certain length depending on the characteristics of the hidden signal. In this paper for the sake of simplicity we use rectangular windows, although other types of windows might in principle be possible and suitable and will not be excluded from the general model. Based on the markers and a predefined BR rule R^{BR} the signal $O(t)$ is cut into instances, i.e., windows $W_x^{m(t)}$ defined as

$$W_x^{m(t)} = \{O(t) \mid R_{low}^{m(t)} < t < R_{up}^{m(t)}\}, \tag{5}$$

where x is the number of windows chosen for one marker type m at a certain time point i, $R_{low}^{m(t)}$ for the start of the window and $R_{up}^{m(t)}$ for the end. For a certain marker $m(i)$ several windows can be cut. These windows can overlap over a certain time period, start and end before or after the time $t = i$. Different windows of different types $W' \subset W$, where W is the space of windows, may be of interest during training or test of eBR as defined by the BR rule R^{BR}.

2.4 Signal Processing, Classification and Post-processing (SP)

During the last step of eBR all relevant windows are processed within a signal processing chain: $SP_f \circ SP_{f-1}...SP_1$. This transfers $W' \subset W$ to Y as output of signal processing and classification. The output for processing a relevant $W^{m(i)}$ is $y_i \in Y$. Each output y_i can be correlated to a prediction score for the likelihood of a brain state that was present in the past, since y_i is only available at time $t = k$, with

[2] see video "Recognition of warnings during teleoperation" referenced in Section Supporting Media

$$k = R_{up}^{m(i)} + j, \text{for } R_{up}^{m(i)} > i \tag{6}$$

or

$$k = i + j, \text{for } R_{up}^{m(i)} \leq i,$$

with j is the time required for all steps of SP.

2.5 Inference of Behaviour (PB)

The mapping between the output $y_i \in Y$ as the likelihood for a certain brain state $s_i \in S$ as defined by the BR rule R^{BR} and the likelihood of future behaviour $b_q \in B$ which may require an adaptation of the HMI is defined by the adaptation rule R^{AHMI}. Based on the output y_i a possible *future* behaviour (b_q) at time $t = q$, with $q > k$, can thus be inferred to adapt the HMI *before* the future behaviour is expressed.

The prediction time p, defined as $p = q - k$, must thus be positive ($p > 0$). Further, to adapt the HMI at $t = p$ for the future behaviour some adaptation time r is required which depends on the adapted system and its control mechanism. To adapt an HMI early enough $p > r$ must be fulfilled. Thus, it is not enough to infer a certain behaviour b_q before it is executed but it must be predicted early enough to enable the adaptation of the HMI before the inferred behaviour b_q is expressed at time $t = q$. Hence, an effective prediction time p_e is required which can be defined as

$$p_e = p - r, \tag{7}$$

with $p_e > 0$.

2.6 Adaptation of the HMI (A)

Adaptation requires a certain time r and takes place between time $t = k$ and $t = k+r$. The mapping between S and B and the kind of adaptation is defined by the adaptation rule R^{AHMI}. The kind and strength of adaptation may not only depend on the inferred behaviour B but also on the output of systems that control the adaptation, e.g., the HMI itself or other supportive systems (Fig. 3).

2.7 Control and Correction of the Adaptation of an HMI (C)

Since BR analysis and thus the detection of the brain state S cannot always be correct and also the mapping between the detected brain state S and the inferred behaviour B might be wrong or contain uncertainties (i.e., a behaviour $b_q^* \in B^*$ might be executed that was not inferred ($b_q^* \neq b_q$)), it is important to implement control mechanisms into the eBR approach that either correct false adaptations of the HMI or prevent them.

To correct inappropriate adaptation, the expressed behaviour b_q^* of the human has to be monitored and compared with the predicted behaviour b_q to search for discrepancies. If such discrepancies are detected the HMI can be adapted again (or readapted) to better meet the detected behaviour b_q^*. Should a wrong adaptation of the HMI possibly result in a malfunction of the HMI, it is not useful to implement control mechanisms that analyse the discrepancy between a predicted behaviour b_q and the expressed behaviour b_q^*. In this case, it is better to combine the prediction made based on brain activity analysis with predictions made based on other measures like the EMG. By implementing automated control and correction, a malfunction of the whole system in case of misclassification of the brain state S by BR or falsely inferred behaviour B is avoided.

3 Evaluation of the Model in Real Applications

In the following we will evaluate the developed formal model based on two implementations that are first described in a more descriptive fashion to point out relevant parts for eBR and then in a formal way with applied rules. We show on parts of both implementations that the formal model contributes a detailed description (of those parts) that not only enhances reproducibility of each individual implementation but also improves comparability between implementations. The later is shown by actually comparing parts of the implemented formal model for both implementations (see Section 3.1). We show that even small differences, e.g., in choice of training or testing instances, can be covered by the formal model for eBR. Further, we show that the formalisation of complex applications is not only a prerequisite to apply verification methods but does already help to optimise procedures and to avoid or ease the finding of errors that are otherwise invisible or at least very hard to be found (see Section 3.2).

Fig. 2 depicts a real robotic application for telemanipulation and telepresence[3]. In this scenario two different implementations for eBR are applied to adapt: (i) an exoskeleton in the implementation *AExo* and (ii) an operator monitoring system (OMS) in the implementation *AOMS*. In (i) BR detects brain states that are related to movement preparation. In (ii) it detects brain states that are related to target recognition, i.e., recognition of important information. The detected brain states are then used to infer on upcoming behaviour: In (i) on the onset of self-initiates arm movements and in (ii) on response behaviour of the subject to the given task. For both inferred upcoming behaviours the interfaces are adapted: In (i) the exoskeleton is adapted to react faster on inferred movement onset and in (ii) the OMS is adapted to permit a longer response time in case that response behaviour is inferred.

In the following we define a formal description for both implementations. To improve the behaviour of an exoskeleton during telemanipulation[4] in the implementation *AExo* (Fig. 2) eBR is used to detect the brain state of movement preparation S^{MP} and no movement preparation S^{noMP} to prepare the HMI, i.e., exoskeleton, for the execution of self-induced movements B^{*MO} in case of S^{MP}. Assumptions about the brain states S are made based on the output Y of a trained classifier as result of the analysis of brain activity as defined in the BR rule R^{Mov}. Adaptation of the exoskeleton's control are made based on a mapping between Y and the likelihood of an inferred upcoming behaviour b^*_q with $b^*_q \in B^{*MO}$ as defined in the rule for adapting the HMI R^{AExo}. In this implementation eBR does reduce the effort, i.e., required force, the user has to invest to lockout the system from a rest position and by this allows a smoother and more intuitive interaction (Folgheraiter *et al.*, 2012). In short, the more likely the upcoming behaviour b^*_q is, the shorter a subject will have to press against force sensors that are integrated in the exoskeleton to release the exoskeleton from a supporting rest position in which it keeps the operator's arm in a fixed position.

To improve the operators support by an OMS[5] (Fig. 1 and 2) in *AOMS* we implemented an approach that automatically analyses the operators brain state to predict whether he recognised important information (i.e., whether he is in the brain state of S^{CPerc}) and will thus likely respond to them (show response behaviour B^{*CPerc}) or whether he did not recognise important information (i.e., is in the brain state $S^{noCPerc}$) and will likely not respond. The rules for the predictions of both possible brain states are defined in the BR rule R^{Perc}. Made predictions are used to adapt the OMS with respect to the tolerated response time, i.e., only a short response time is allowed in case that no response is predicted and a long response time is allowed in case a response is predicted as defined in the adaptation rule R^{AOMS} (Kirchner *et al.*, 2010).

3.1 Coverage of Differences in Implementations by the Model

When formally comparing the implementations *AExo* and *AOMS* important differences exist with respect to, e.g. the windowing procedures, the rules for adding makers by BR and the type of performed control and correction. These differences must be considered and become very important when both implementations should be integrated in one flow, i.e., to adapt the OMS and the exoskeleton within one application and by one eBR flow as it is the case in the scenario displayed in Fig. 2. On the other hand, we will show that despite those differences the general model does fit both implementations.

In both implementations the analog signal $A \in \mathbb{R}^{124}(t)$ was recorded with 124 channels. After digitisation with 16 bit it was sampled with $f = 5000\,\text{Hz}$ as defined in the BR rules R^{Perc} and R^{Mov}.

[3]see video "Telecontrol scenario" referenced in Section Supporting Media
[4]see also video "Movement prediction for exoskeleton control" referenced in Section Supporting Media
[5]see video "Recognition of warnings during teleoperation" referenced in Section Supporting Media

Table 1: Implementations of part WS (Fig. 3) of the model for training of eBR.

<table>
<tr><td>AdpatExo (WS in training)</td><td>AOMS (WS in training)</td></tr>
<tr><td>$W_x^{m(t)}$ as defined in BR rule R^{Mov}:</td><td>$W_x^{m(t)}$ as defined in BR rule R^{Perc}:</td></tr>
<tr><td colspan="2">windows for training of eBR: In AExo windows are chosen before windowing, since movement preparation S^{MP} happens before movement onset (Kornhuber and Deecke, 1965; Balconi, 2009). Same is true for instances of type S^{Ign} in AOMS. However, instances for S^{CPerc} have to be chosen after windowing since the choice of window depends on the response of the operator.</td></tr>
<tr><td>Instances from the rest period are chosen to train for the class S^{noMP}:

Only markers of type m_{noMP} are considered that occur within the rest period between the markers m_{offset} and m_{onset} in case there is no other marker 2000 ms before and after m_{noMP}.

$W_1^{m_{noMP}} = \{O(t) \mid (i - 1000\,ms) < t < i\}$, with $m(i) = m_{noMP}$.

To select instances of the brain state S^{MP}, two windows within each rest period with respect to one m_{onset} marker are chosen:

$W_1^{m_{onset}} = \{O(t) \mid (i - 950\,ms) < t < (i + 50\,ms)\}$, with $R_{low}^{m_{onset}} = i - 950\,ms$ and $R_{up}^{m_{onset}} = i + 50\,ms$.

$W_2^{m_{onset}} = \{O(t) \mid (i - 1100\,ms) < t < (i - 100\,ms)\}$, with $R_{low}^{m_{onset}} = i - 1100\,ms$ and $R_{up}^{m_{onset}} = i - 100\,ms$.</td><td>Instances of the class of behaviour B^{*CPerc} are defined as:

$W_1^{m_{resp_x}} = \{O(t) \mid i\}$, with $m(i) = m_{resp_1}$, or $m(i) = m_{resp_2}$, $R_{low}^{m_{resp_x}} = i\,ms$, and $R_{up}^{m_{resp_x}} = i + 1\,ms$.

$W_1^{m_{rel_x}} = \{O(t) \mid i\}$, with $m(i) = m_{rel_1}$, or $m(i) = m_{rel_2}$, $R_{low}^{m_{rel_x}} = i\,ms$, and $R_{up}^{m_{rel_x}} = i + 1000\,ms$ are chosen for the class S^{CPerc}, if followed by $W_1^{m_{resp_x}} = \{O(t) \mid i\}$. The rule makes sure that after an instance ${s_i}^{CPerc}$ an instance of type B^{*CPerc} follows (the target was perceived).

$W_1^{m_{irr}} = \{O(t) \mid i\}$, with $m(i) = m_{irr}$, $R_{low}^{m_{irr}} = i\,ms$, and $R_{up}^{m_{irr}} = i + 1000\,ms$ are chosen for the class S^{Ign} in case there is no marker of type m_{resp_x} or m_{rel_x} 2000 ms before or after a marker m_{irr}.</td></tr>
</table>

The output of analog-digital conversion is defined as

$$o(i) = \begin{bmatrix} d_1(i) \\ d_2(i) \\ . \\ . \\ d_{124}(i) \end{bmatrix}, \tag{8}$$

with $d_j \in \mathbb{N}', \mathbb{N}' \subset \mathbb{N}$ and $\mathbb{N}' = \{-2^{15}, -2^{15}+1, ..., 2^{15}\}$.

Further, in both implementations markers are defined as

$$M^{AOMS} = M^{AExo} = \{-2^{15}, -2^{15}+1, ..., 2^{15}\} \tag{9}$$

and $m(t) = -1$ for *no marker*.

Here the main difference between both implementations is that markers for training of BR were generated by different systems as allowed by the formal model (Fig. 3). In *AOMS* the HMI, the eBR system itself, and a position tracking system (PTS) as a supportive system generate the required markers, whereas in *AOMS* only the HMI generates markers.

For the implementation of *AOMS* it was important to understand the rules for choosing training examples (rules are given in Table 1). The formalisation of eBR was used here to better understand the procedures. In this particular application (Fig. 2), it was not possible to generate enough training examples for the class $S^{noCPerc}$. Instead of using methods that can cope with few training examples (Fazli *et al.*, 2009; Lotte and Guan, 2010) we decided to substitute the examples of the underrepresented class with examples that were expected to evoke similar brain activity to achieve a higher prediction performance (Kirchner *et al.*, submitted). Threshold adaptation was developed to later cope with the fact that the built classifier is not optimal to classify in the test case (Metzen and Kirchner, 2011). In *AOMS*, brain activity present during the brain state S^{Ign} that is evoked by ignored, unimportant stimuli (labeled by m_{irr}) was expected to be similar to brain activity present during the brain state $S^{noCPerc}$ that was evoked by important stimuli (labeled by m_{rel}) to which the user did not respond (see Table 1). Thus, windows $W_x^{m(t)}$ as defined in BR rule R^{Perc} with $W_1^{m_{irr}}$ were chosen to train for the brain state $S^{noCPerc}$, while windows defined by $W_1^{m_{rel}}$ were chosen to train for the brain state S^{CPerc} in case there was a response on the important warning, i.e., $W_1^{m_{rel}}$ was followed by an instance of type $W_1^{m_{resp}}$.

While instances for the training classes S^{CPerc} in *AOMS* were, as described before, chosen *after* windowing just by defining an *order of instances*, instances for the class S^{MP} in *AExo* were *not* chosen *after* windowing but with respect to their distance in time to the marker m_{onset} *during* windowing (see Table 1 for more details). The marker m_{onset} labels the time point of movement onset of the user's arm after a rest period. Note that two windows are defined for each m_{onset} (Kirchner *et al.*, 2013). Although both implementations show differences in the procedure for choosing relevant instances (windows), both implementations are covered by the model (Figure 3). This shows that our model is general enough to cover differences in implementation, while still allowing a detailed description of relevant parts of the implementation.

More examples for the capability of the model to cover differences in the implemented procedures can be given for the test phase, i.e., application of eBR. The main difference between both implementations during test is, that in *AOMS* it is known when to classify EEG instances $W_1^{m_{rel}}$, since only after an important information (warning) is presented to the user a classification of the brain state (S^{CPerc} versus $S^{noCPerc}$) is required. On the other hand in *AExo* it is *not* known at what time classification by BR is important, since it is unknown at what time the operator wants to start to move (to end a rest period). Here, instances $W_1^{m_{BRwin}}$ are therefore cut continuously, i.e., every 50 ms (Table 2) based on the marker m_{BRwin} that is automatically added by the BR system to label the end of each instance.

For adapting the exoskeleton in *AExo*, the adaptation time r is more relevant than for adapting the OMS in *AOMS* and continuous prediction of relevant behaviour is only required in *AExo* (Table 3). Here, the BR continuously provides values for Y and sends them to the HMI while the exoskeleton makes only use of an output y_i in case it was locked in for some time ($4\,s$ in this implementation), i.e., is in a rest period. Thus, during test the HMI does control its adaptation as defined in the

Table 2: Implementations of part WS (Fig. 3) of the model for test of eBR.

<table>
<tr><td>AdpatExo (WS in test)</td><td>AOMS (WS in test)</td></tr>
<tr><td>$W_x^{m(t)}$ is defined in BR rule R^{Mov}.</td><td>$W_x^{m(t)}$ is defined in BR rule R^{Perc}.</td></tr>
<tr><td colspan="2">windows for test of eBR: In AExo windows are chosen independent of the state of the HMI. In AOMS they are chosen with respect to the state of the HMI, i.e., in case that a warning was presented.</td></tr>
<tr><td>Instances of class S^{noMP} or
S^{MP} are defined as:

$W_1^{m_{BRwin}} =$
$\{O(t) \mid (i - 1000\,ms) < t < i\}$,
with $m(i) = m_{BRwin}$,
$R_{low}^{m_{onset}} = i - 1000\,ms$,
and $R_{up}^{m_{onset}} = i$.</td><td>Instances of class S^{CPerc} or
or $S^{noCPerc}$ are defined as:

$W_1^{m_{rel_x}} = \{O(t) \mid i\}$,
with $m(i) = m_{rel_1}$, or
$m(i) = m_{rel_2}$, $R_{low}^{m_{rel_x}} = i\,ms$,
and $R_{up}^{m_{rel_x}} = i + 1000\,ms$.</td></tr>
</table>

model but has no influence on the eBR system, while the OMS in *AOMS* is actively requesting predictions from the eBR system after warnings and is thus controlling both the eBR system and its own adaptation. Finally, the HMI does control correction procedures. For both implementations it controls whether the inferred behaviour, B^{MO} or B^{CPerc} is actually executed, i.e., $B = B^*$. However, in *AExo*, the behaviour of the HMI is only changed in case the execution of it is detected (see explanations given in Section 1), while in *AOMS* the behaviour of the OMS is changed right away with respect to the outcome of BR by, e.g., extending the allowed response time in case that B^{CPerc} is predicted (Table 3). This adaptation is then only controlled afterwards by monitoring the actual response behaviour B^* of the user by the OMS. Hence, in case that B^{CPerc} was predicted but $B^{*noCPerc}$ ($B \neq B^*$) is detected by the HMI within the extended allowed response time a second warning is presented.

In summery, we showed that for the given implementation examples different procedures are applied for the generation of markers, for choosing training windows, for the choice of relevant situations that require BR and for the adaptation of the HMIs by eBR. To allow this, different parts of the model are differently implemented. Despite those differences in implementation of the model the general formal model can be applied for both implementation examples and can be used to formalise the different parts and to compare for differences.

3.2 Detection of Implementation Errors by Formalisation

The application of the formal model does not only allow to compare different implementations but enables the detection of implementation errors within an individual implementation of the formal model. In the following we give two examples for errors that could only be detected by formalising both implementations. Both errors were due to the implemented control mechanism for eBR (C in Fig. 3) ”invisible” and would not lead to faulty behaviour of the system but would reduce performance and adequacy in the adaptation of the HMI. Hence, supervising the correctness of the total systems behaviour would not allow to uncover the here described implementation errors.

By formalising the implementation *AExo* with respect to the general model of eBR we could uncover an implementation error that was caused by misinterpreting the outcome Y of SP for the adaptation (A) of the HMI (Fig. 3). On the exoskeletons control side it was expected that in the case of no movement preparation no value, i.e., $y_i = 0$ should be the output of BR. However, Table 3 shows that

Table 3: Implementations of part A (Fig. 3) of the model for adapting the HMI.

AdpatExo (A)	AOMS (A)
The output Y modulates the time threshold T_{th} of the force sensors in the exoskeleton, i.e., in case of B^{CRes} the user has to press shorter against sensors to lockout the system from rest while executing B^{*CRes} as defined in the adaptation rule R^{AExo}.	The output Y modulates the allowed response time (RT) for the user that is controlled by the OMS as defined in R^{AOMS}.
$T_{th}(k) = (T_{th}^{Max} - (1 - 2(y - 0.5)) + T_{th}^{Min}$ with $y_i = 1$ for minimal time threshold T_{th}^{Min} and maximum movement prediction impact here: $T_{th}^{Min} = 10\,\mathrm{ms}$, since control frequency is $100\,\mathrm{Hz}$ and with $y_i \leq 0.5$ for maximal time threshold T_{th}^{Max} experimentally determined to avoid unwanted lockout.	For $y_i = 1$ the allowed RT is increased from $2\,\mathrm{s}$ to $RT_{max} = 10\,\mathrm{s}$ for $y_i = -1$ *no* adaptation of RT takes place ($RT_{min} = 2\,\mathrm{s}$).
The time that is required for each adaptation r depends on the frequency of predictions made by BR (here every $50\,\mathrm{ms}$) and the time required to adapt the time threshold T_{th} (here $10\,\mathrm{ms}$). The adaptation time r for a certain time point i is in the worst case $r_i = 60\,\mathrm{ms}$. An effective prediction $p_e = (q - k) - r$ can be as early as $190\,\mathrm{ms}$ before an movement onset ${b_q}^{*MO}$ since a high classification performance can be achieved at $k = q - 250\,\mathrm{ms}$, with $q = 0\,\mathrm{ms}$ (Folgheraiter *et al.*, 2011) before B^{*MO}.	The time that is required to adapt RT r takes $r \approx 12\dot{m}\mathrm{s}$ and can be neglected since adaptation is only required before RT_{min} reduced by $R_{up}^{m_{rel}}$ and j. The OMS requests predictions of the brain state by eBR only after the presentation of important information (targets) to adapt RT if ${b_q}^{*CPerc}$ is not executed before RT_{min} The HMI controls the eBR system and its own adaptation.

eBR was not just predicting the brain state of movement preparation S^{MP} (in case of $y_i > 0.5$) but also the brain state of *no* movement preparation S^{noMP} (in case of $y_i \leq 0.5$), which is not relevant for this application. In this faulty implementation the exoskeleton would erroneously have been adapted for a faster lockout in case of $y_i \leq 0.5$ although *no* movement preparation was predicted by BR. This error would not have changed the total behaviour of the exoskeleton but would have resulted in an inadequate adaptation of the exoskeleton and possibly reduced comport for the user.

In the implementation *OMS* an error was found within the implementation of the part SP of the model (Fig. 3). Here, training of eBR should take place on instances of type $W_x^{m_{rel_1}}$ and $W_x^{m_{rel_2}}$ (Table 1). Instances of type $W_x^{m_{rel_1}}$ are *first* warnings and instances of type $W_x^{m_{rel_2}}$ are repeated *second* warnings that are visually highlighted (by changing the colour). During test BR does analyse the brain state after both types of warnings to enable inference of response behaviour by eBR. However, in the setting a *third* type of warning was used ($W_x^{m_{rel_3}}$). This *third* warning was very strong. We expected that the user would in all cases respond to this warning, especially since the control of the robot was removed from the user after the warning was shown for 1000 ms. Hence, it was not required to detect the brain state after the presentation of the *third* warning. Further, EEG patterns evoked by the *third* warnings (type $W_x^{m_{rel_3}}$) were quite different compared to EEG patterns evoked by the warnings of type $W_x^{m_{rel_1}}$ and type $W_x^{m_{rel_2}}$. Thus, by training the classifier on instances of type $W_x^{m_{rel_3}}$ performance in the classification of the brain state for instances of type $W_x^{m_{rel_1}}$ and $W_x^{m_{rel_2}}$ might have dropped. By formalising this implementation the error was found.

Both examples show that the developed formal model for eBR enabled the detection of errors in implementations and that a clear definition of rules and the kind of outcome of subsystems is important for the overall functionality of the whole system.

4 Conclusion and Outlook

In the presented work we developed for the first time a general, formal model for eBR, which was developed by our group for the *safe* adaptation of interfaces in robotic applications using unreliable data, i.e., EEG data, to allow a better support of inferred upcoming interaction behaviour. The developed formal model does focus on the interaction between subsystems, thus it does not describe how exactly each subsystem with respect to its functionality or the individual data processing, e.g., kind of signal processing and classification methods, is implemented, but describes what kind of data and information is used, exchanged and transferred between subsystems. We showed and discussed on implementation examples that (1) the developed formal model fits different implementations, (2) covers differences in the implemented procedures of different parts of the model, and (3) could uncover errors that are difficult or not at all to find without formalisation. A formal model for complex systems, as presented here, enables a very detailed as well as clear description of procedures. This becomes more important as more interdisciplinary research is required to develop complex systems for advanced human-machine interaction, since it can ease their implementation for different applications as explained using examples.

We further want to state that the formalisation of eBR is important for the introduction of this approach in practical use cases and industry, since it allows to *verify* that *eBR functions error-free* while being adaptive to different requirements. One application of growing interest is robotic-based rehabilitation. To apply robotic systems for rehabilitation, it has to be assured that a developed system works correctly and is thus safe to be applied on human. The challenge is to guarantee that such systems are correct and complete when developed (Drechsler *et al.*, 2012) and work error-free while being adaptive to different requirements of different groups of patients and their state in rehabilitation (Kirchner *et al.*, 2013). Alternatively, also approaches can be studies that start with a formal specification, like Event-B (see Abrial *et al.*, 2010).

However, not only in medical applications more intense and intuitive interaction between humans and technical systems is required, but also new paths have to be followed for improving interaction in different application fields of industry. These new and advanced approaches should allow real cooperation between technical systems and human, be it for the general improvement of workflows, to allow a technical system to make use of human cognitive resources or for an individually and situation specific support of elderly employees in, e.g., heavy physical work. To allow such advanced human-machine interaction, technical systems must understand the human's intention. Its interpretation might not be free of errors, as it is also true when human interpret human's intentions.

We showed that eBR thanks to the implemented automated control mechanism allows to deal with possibly ambiguous interpretation of the humans intention. However, online monitoring of the correctness and completeness of complex systems' behaviour and interaction behaviour is required to apply eBR in industry and for medical purposes. To establish such monitoring formal models for complex systems constitute the main precondition to apply verification tools on system level and can already help to better understand such complex systems and intra-system dependencies as well as to detect hidden errors as shown in this work.

Supporting Media

As supporting media videos are available that show the "Telecontrol scenario" (http://youtu.be/8YHhjQiv6JE) depicted in Fig. 2 with both implementations of eBR: "Movement prediction for exoskeleton control" (http://youtu.be/fiI4MKPTFg0) and "Recognition of warnings during teleoperation" (http://youtu.be/8WEVZz6bpJU).

Acknowledgment

This work was supported in part by the German Ministry of Economics and Technology (grant no. 50 RA 1011 and grant no. 50 RA 1012) and the German Research Foundation (DFG) within the Reinhart Koselleck project under contract no. DR 287/23-1.

References

Abrial, J.-R., Butler, M. J., Hallerstede, S., Hoang, T. S., Mehta, F. and Voisin, L. (2010), "Rodin: an open toolset for modelling and reasoning in Event-B", *International Journal on Software Tools for Technology Transfere (STTT) - Special Session on VSTTE 2008*, Vol. 12 No. 6, pp. 447–466.

Balconi, E. (2009), "The multicomponential nature of movement-related cortical potentials: functional generators and psychological factors", *Neurophysiological Trends*, Vol. 5, pp. 59–84.

Coles, M. (1989), "Modern mind-brain reading: Psychophysiology, physiology, and cognition", *Psychophysiology*, Vol. 26 No. 3, pp. 251–269.

Drechsler, R., Diepenbeck, M., Große, D., Kühne, U., Le, H. M., Seiter, J., Soeken, M. and Wille, R. (2012), "Completeness-driven development", *International Conference on Graph Transformation, LNCS 7562*, Bremen, Germany, pp. 38–50.

Drechsler, R. and Große, D. (2005), "System level validation using formal techniques", *IEE Proceedings Computer & Digital Techniques, Special Issue on Embedded Microelectronic Systems: Status and Trends*, Vol. 152 No. 3, pp. 393–406.

Fazli, S., Popescu, F., Danóczy, M., Blankertz, B., Müller, K. and Grozea, C. (2009), "Subject-independent mental state classification in single trials", *Neural Networks*, Vol. 22 No. 9, pp. 1305–1312.

Ferrez, P. W. and Millán, J. d. R. (2008), "Error-related EEG potentials generated during simulated brain-computer interaction", *IEEE Transaction on Biomedical Engineering*, Vol. 55 No. 3, pp. 923–929.

Folgheraiter, M., Jordan, M., Straube, S., Seeland, A., Kim, S. K. and Kirchner, E. A. (2012), "Measuring the improvement of the interaction comfort of a wearable exoskeleton", *International Journal of Social Robotics*, Vol. 4 No. 3, pp. 285–302.

Folgheraiter, M., Kirchner, E. A., Seeland, A., Kim, S. K., Jordan, M., Wöhrle, H., Bongardt, B., Schmidt, S., Albiez, J. and Kirchner, F. (2011), "A multimodal brain-arm interface for operation of complex robotic systems and upper limb motor recovery", in *P. Vieira, A. Fred, J. Filipe and H. Gamboa, eds, Proceedings of the 4th International Conference on Biomedical Electronics and Devices (BIODEVICES-11)*, SciTePress, Rome, Italy, pp. 150–162.

Haynes, J.-D. and Rees, G. (2006), "Decoding mental states from brain activity in humans", *Nature Reviews Neuroscience*, Vol. 7 No. 7, pp. 523–534.

Hodges, P. W. and Bui, B. H. (1996), “A comparison of computer-based methods for the determination of onset of muscle contraction using electromyography.”, *Electroencephalogr Clin Neurophysiol*, Vol. 101 No. 6, pp. 511–519.

Kamitani, Y. and Tong, F. (2005), “Decoding the visual and subjective contents of the human brain.”, *Nature Neuroscience*, Vol. 8 No. 5, pp. 679–685.

Kirchner, E. A., Albiez, J., Seeland, A., Jordan, M. and Kirchner, F. (2013), “Towards assistive robotics for home rehabilitation”, in *M. F. Chimeno, J. Solé-Casals, A. Fred and H. Gamboa, eds, Proceedings of the 6th International Conference on Biomedical Electronics and Devices (BIODEVICES-13)*, SciTePress, Barcelona, Spain, pp. 168–177.

Kirchner, E. A., Kim, S. K., Straube, S., Seeland, A., Wöhrle, H., Krell, M. M., Tabie, M. and Fahle, M. (submitted), ‘On the applicability of brain reading for self-controlled, predictive human-machine interfaces in robotics’.

Kirchner, E. A. and Tabie, M. (2013), “Closing the gap: Combined EEG and EMG analysis for early movement prediction in exoskeleton based rehabilitation”, *Proceedings of the 4th European Conference on Technically Assisted Rehabilitation (TAR 2013), session 4, event 1*, Berlin, Germany, pp. 1–4.

Kirchner, E. A., Wöhrle, H., Bergatt, C., Kim, S. K., Metzen, J. H., Feess, D. and Kirchner, F. (2010), “Towards operator monitoring via brain reading – an EEG-based approach for space applications”, *Proceedings of the 10th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Sapporo, Japan, pp. 448–455.

Kornhuber, H. H. and Deecke, L. (1965), “Hirnpotentialänderungen bei Willkürbewegungen und passiven Bewegungen des Menschen: Bereitschaftspotential und reafferente Potentiale”, *Pflügers Archives*, Vol. 284, pp. 1–17.

Lotte, F. and Guan, C. (2010), “Learning from other subjects helps reducing Brain-Computer interface calibration time”, *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, ICASSP ’10, Dallas, Texas, USA, pp. 614–617.

Metzen, J. H. and Kirchner, E. A. (2011), “Rapid adaptation of brain reading interfaces based on threshold adjustment”, *Proceedings of the 2011 Conference of the German Classification Society, (GfKl-2011)*, Frankfurt, Germany, p. 138.

Miyawaki, Y., Uchida, H., Yamashita, O., Sato, M.-a., Morito, Y., Tanabe, H. C., Sadato, N. and Kamitani, Y. (2008), “Visual Image Reconstruction from Human Brain Activity using a Combination of Multiscale Local Image Decoders”, *Neuron*, Vol. 60 No. 5, pp. 915–929.

Naselaris, T., Prenger, R. J., Kay, K. N., Oliver, M. and Gallant, J. L. (2009), “Bayesian Reconstruction of Natural Images from Human Brain Activity”, *Neuron*, Vol. 63 No. 6, pp. 902–915.

Polyn, S. M., Natu, V. S., Cohen, J. D. and Norman, K. A. (2005), “Category-specific cortical activity precedes retrieval during memory search.”, *Science (New York, N.Y.)*, Vol. 310 No. 5756, pp. 1963–1966.

Seno, B. D., Matteucci, M. and Mainardi, L. (2010), “Online detection of P300 and error potentials in a BCI speller”, *Computational Intelligence and Neuroscience*, Vol. 2010, pp. 1–5.

Suppes, P., Perreau-Guimaraes, M. and Wong, D. K. (2009), “Partial orders of similarity differences invariant between EEG-recorded brain and perceptual representations of language”, *Neural Computation*, Vol. 21 No. 11, pp. 3228–3269.

Tabakov, D., Vardi, M. Y., Kamhi, G. and Singerman, E. (2008), “A temporal language for systemc”, *Proceedings of the 2008 International Conference on Formal Methods in Computer-Aided Design*, FMCAD ’08, IEEE Press, Portland, Oregon, USA, pp. 1– 9.

Tabie, M. and Kirchner, E. A. (2013), “EMG onset detection - comparison of different methods for a movement prediction task based on EMG”, in *S. Alvarez, J. Solé-Casals, A. Fred and H. Gamboa, eds, In Proceedings of the 6th International Conference on Bio-inspired Systems and Signal Processing (BIOSIGNALS-13)*, SciTePress, Barcelona, Spain, pp. 242–247.

Wöhrle, H., Teiwes, J., Kirchner, E. A. and Kirchner, F. (2013), “A framework for high performance embedded signal processing and classification of psychophysiological data”, *APCBEE Procedia. International Conference on Biomedical Engineering and Technology (ICBET-2013)*, Elsevier, Kopenhagen, Denmark.

Zander, T. O. and Kothe, C. (2011), "Towards passive brain-computer interfaces: applying brain-computer interface technology to human-machine systems in general", *Journal of Neural Engineering*, Vol. 8 No. 2, pp. 025005.

Figures

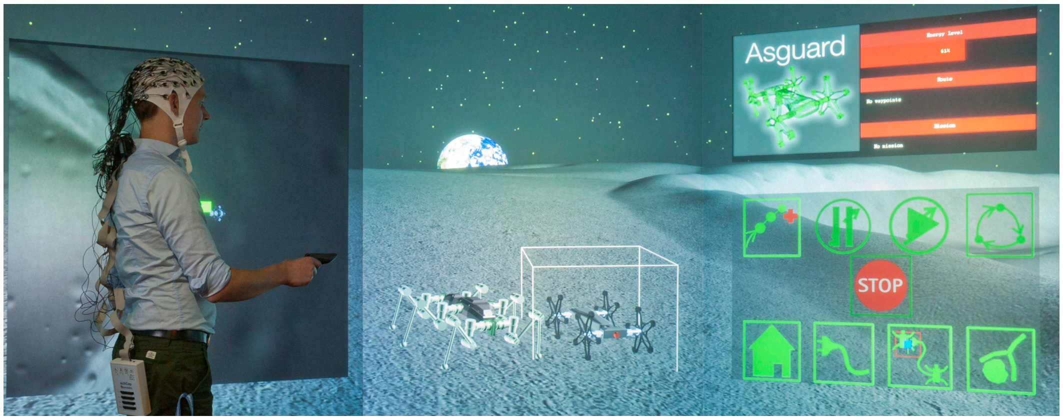

Figure 1: Multi-robot control supported by embedded brain reading.

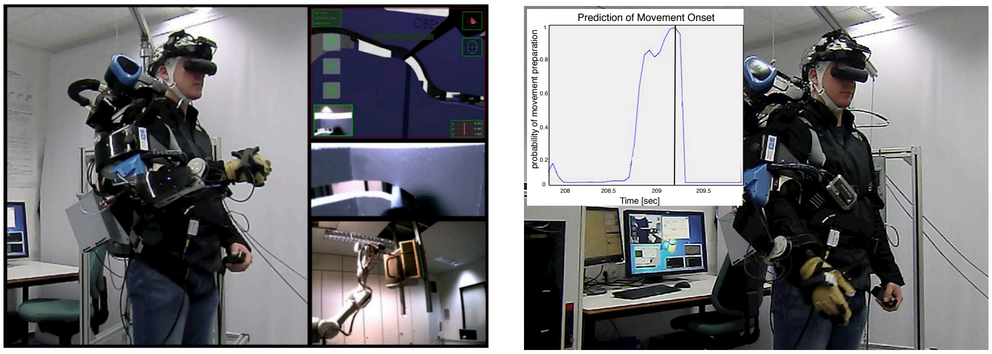

Figure 2: Left: Telecontrol of a real robotic arm through a labyrinth by means of an exoskeleton and simultaneous responses to warnings by the operator: embedded brain reading adapts exoskeleton control and operator monitoring system. Right: Prediction of movement preparation in single trial (blue line) before movement onset (black vertical line) to support movement onset. Time (x-axis) shows absolute time from the start of the experiment to a chosen single example of BR output.

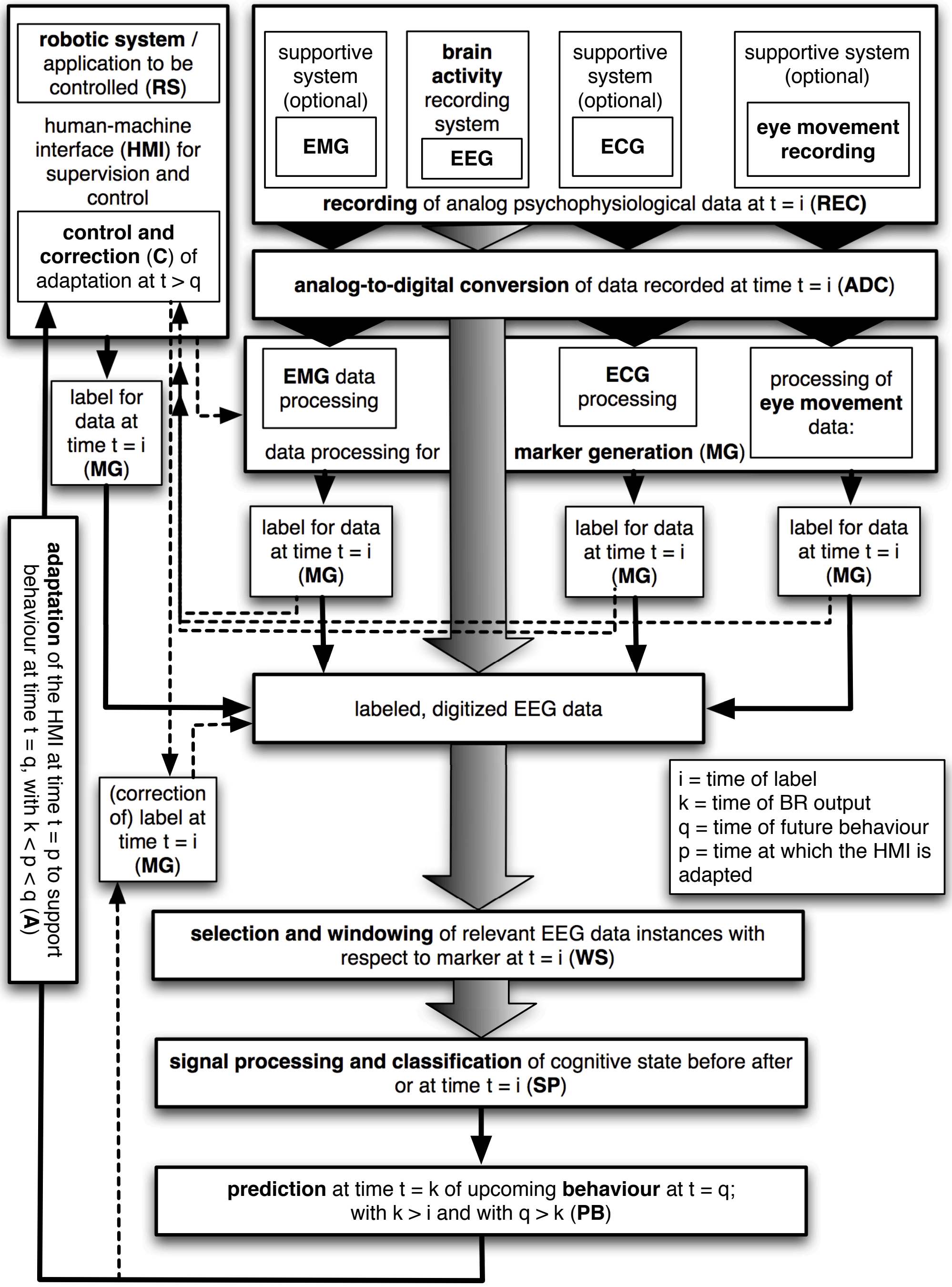

Figure 3: Model for embedded brain reading in a formal structured form.