29

Standard model, file formats and methods in Brain-Computer Interface research: why?

Lucia Rita Quitadamo[1,2], Donatella Mattia[2], Febo Cincotti[2], Fabio Babiloni[3], Gian Carlo Cardarilli[4], Maria Grazia Marciani[1,2] and Luigi Bianchi[1,2,5]

[1]*University of Tor Vergata, Department of Neuroscience*
[2]*Fondazione Santa Lucia, IRCCS, Neuroelectrical Imaging and BCI Laboratory*
[3]*University of La Sapienza, Department of Physiology and Pharmacology*
[4]*University of Tor Vergata, Department of Electronic Engineering*
[5]*University of Tor Vergata, Centro di Biomedicina Spaziale*
[1,2,3,4,5]*Rome, Italy*

1. Introduction

Assistive Technologies (AT) include all the assistive, adaptive and rehabilitation devices that help people with disabilities to interact with the external environment without or with minimal need of care assistant. They have become very wide spread in the last decades as they promote independence for those people that are unable to perform a task, by providing them methods of interacting with the technology needed to accomplish such tasks.
Brain-Computer Interface (BCI) represents a subset of the more general AT; its main purpose is to help disables to communicate by creating a direct channel of interaction between brain and external environment, without the need of muscles or nerves (Wolpaw et al., 2002; Kübler & Neumann, 2005): in fact, people who lost the control on their muscles, after strokes, spinal cord injuries, cerebral palsy, traumas or degenerative diseases (see Amyotrophic Lateral Sclerosis), may lie in the so-called "locked-in" state, that is, they are confined into a body which does not meet their intents and desires anymore, while their cognitive activity is still intact; these people can keep on communicating their thoughts by means of a BCI, which translates their brain signals into output controls that can be commands to select characters on a speller or to pilot a wheelchair or a robotic arm, as well as commands to control a cursor on a screen or a domotic environment and so on.
It is evident that the BCI field is a very complex one, as it deals either with human beings feelings and technology aspects; in fact BCI must involve different research areas such as engineering, informatics, computer science, neurology, neurophysiology, psychology, rehabilitation, that must interact to implement an efficient and user-centric BCI system.
It follows that one of the greatest problem that can be found in BCI research is a difficulty in the communication among the people who are involved in it: in fact, a lot of research labs are interested in BCI all around the world, each of them focusing on some particular aspects of these systems (enhancing the acquisition quality of the signals, improving the communication rate, finding the best algorithms to classify data, choosing the best

peripheral according to the user requirements) and maybe most of them have dealt with the same problems and found different solutions to them, so that a multitude of BCI systems has been implemented to date, which are very different and almost incompatible among each other (Cincotti et al., 2006).
Hence a question arises: is it possible to define a common language for BCI systems that allows all the researches to talk the same language and so to share their knowledge?
This question comes from the necessity of defining **standards** in BCI, that mean a common language for BCI systems and that are fundamental to promote some progresses, such as:

- **Easy data sharing** among different research labs. Today many different file formats are used for storing physiological and BCI data, such as BCI2000 (Schalk et al., 2000), Ascii, GDF (Schlögl, 2006), EDF and EDF+ (Kemp & Olivan, 2003), Matlab, etc, and each of them has its own features. This prevents a practical circulation of data among the different scientific communities and hinders the sharing of knowledge.
- **Unique BCI model.** Different structures and protocols for BCI systems have been implemented to date and, due to the lack of a standard model to describe them, there is often a disagreement even on the names of the basic components which constitute them (for example, *trial, run, session*). This adds some difficulties when different modules of a system have to be exchanged or different systems have to be compared.
- **Reliable comparison among different systems.** Today many different metrics are used to evaluate the performances of a BCI system (bit-rate, Mutual Information, Entropy, characters per second, error rate) and each of them focuses on a different problem (e.g. classifier performance, spelling speed, etc…). This leads to a misunderstanding of the effective behavior of a system and makes it difficult to identify the best one for a specific application.
- **Easy module substitution (SW and HW) without compatibility issues.** Actually each lab has its own systems as regarding recording devices, analysis and optimizing tools, classifiers, etc., which are not compatible each other. This, again, hinders the sharing of tools and results among labs**.**

In the following paragraphs, some solutions to the problems previously mentioned will be provided; a unique model for BCI systems, a new file format, a metric for evaluating their performances and tools for optimizing them will be illustrated. All these features together contribute to the creation of a standard language for talking about BCI, that is necessary for the dissemination of knowledge, for the sharing of data and results and finally for the standardization and unification of BCI systems.

2. Open file formats in BCI: a XML-based proposal

The problem of data sharing has always been very compelling in the neuroscience research field, and mainly in the BCI one, as exchanging data among different researchers and laboratories can lead to the sharing of results and to the dissemination of resources. Unfortunately, the actual situation is that each lab uses its own file format and still needs to convert other labs' files in its own format if wants to use them, with waste of resources and time.
The solution to this problem could be reached by means of a common file format that is flexible, easily accessible and comprehensible by everyone and suitable for storing information about all possible kinds of physiological signals and BCI-related components.

This would be a great achievement in the BCI field as it would allow to overcome all the obstacles due to the fact that each laboratory has its own file format to save data.
For this reason, open standard file formats have been recently proposed (Bianchi et al., 2007b), which can be accessible and modified by everyone, by adding or removing information, without breaking the backward compatibility. This file format is based on the XML (eXensible Markup Language) technology and then called NeuroPhysiological data in XML (NPX). XML has some important features, such as, *extensibility,* that is, data can be added by everyone without altering the overall structure of the document and without breaking the backward compatibility; *portability,* that is, the user can define new tags and attributes for the objects, without special libraries for reading them; *platform-independence,* that is, the technology can be used with different operating systems without any problem; *data-independence,* that is, the content of a file is kept separate from its presentation, so that one can store the content in an XML file only once and then extract and visualize it in the desired format. All these features have made XML the standard technology for the communication world.
The NPX format was successfully used for the storage of electroencephalography (EEG), magnetoencephalography (MEG), electromyography (EMG), electrocardiography (EKG) and event-related potentials (ERP) data; it supports a virtually unlimited number of sensors, events and montages; data can also be stored in various ways with respect to the accuracy (8, 16, 32, 64 bits) and the internal representation (integer, floating point). Sometimes, if a faster access to data is needed, an XML file can be linked to a binary file: for example, if the amount of sampled data is huge (e.g. an EEG recording) they can be stored in an additional distinct binary file, otherwise (e.g. ERP, spectral data) they can be stored in the XML file itself. In both cases the XML file will contain a complete description of the sensors (dynamics, number of bits, gain, coordinates, etc…), events (type, occurrence, etc…), processing, etc….
The XML technology has been also adopted for the storage of BCI experiments parameters, such as feedback rules for sensory motor protocols, virtual keyboard layout for spellers, classifiers performances, etc. (Quitadamo et al., 2007).
All these features show that an XML-based file format can be a valuable solution for BCI data storage and handling; the necessity of a common file format is compelling if a unification of BCI resources is the ultimate goal that standardization can achieve.

3. A unique functional model: structural and temporal characterization

A further step that is necessary for the standardization of BCI systems is to outline a set of common definitions for all the BCI components, which are embedded in a functional model. A wide-accepted functional model has been fully described in the literature (Mason & Birch, 2003; Bianchi et al., 2007a; Quitadamo et al., 2008) and is depicted in Fig. 1.
In this model the two most important functional blocks are the Transducer and the Control Interface. The Transducer is the only module that deals with physiological signals; it includes, in fact, different sub-modules for the acquisition and processing of brain signals:

- **The Collector,** that is the module which deals with the acquisition of brain signals and is usually constituted by sensors. Different signals have been used to implement a BCI, some of them being recorded with a non-invasive modality such

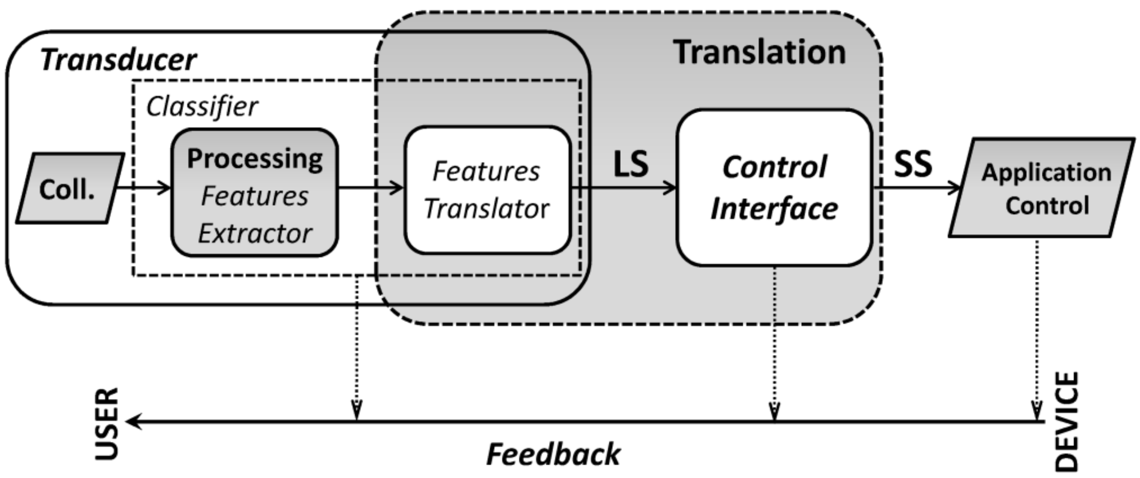

Fig. 1. Functional Model of a BCI system

as, EEG, MEG (Mellinger & al., 2007), functional Magnetic Resonance Imaging (fMRI, Weiskopf et al., 2004; Yoo et al., 2004) and near-infrared spectroscopy (NIRS, Coyle et al., 2007) signals, and some others in an invasive modality, such as electrocorticography (ECoG, Schalk et al. 2008) and intra-electrocortigraphy signals.
For example, in the case of EEG-based BCIs, which are the most diffused ones, scalp signals are recorded by means of a set of electrodes, whose number spans from 2 to 128, according to the application.

- **The Classifier**, that is the module which is devoted to the processing of bioelectrical signals and is formed by the Features Extractor and the Features Translator. The Features Extractor deals with the extraction of some features of interest from brain signals; these features can be manifold, such as P300 (Donchin et al., 2000; Sellers at al., 2006), μ-rhythm (Wolpaw & McFarland, 2004), Slow Cortical Potentials (Kübler et al., 2001), Steady-State Visual Evoked Potentials (Trejo et al., 2006), etc., and lead to the implementation of different BCI protocols. Then these features are linearly or non-linearly combined, by means of the Features Translator, into some Logical Symbols, LSs, which belong to a Logical Alphabet (LA); a LS corresponds to the actual mental task performed by the subject and is usually the result of a classification process. For example, in the fMRI-based protocol implemented by Yoo (Yoo et al., 2004), the subject learnt to control the movement of a cursor through a maze by exploiting four mental tasks: right and left hand motor imagery, mental calculation and mental speech generation. These four tasks have to be recognized by the system, and this corresponds to the generation of a LS (for example, one among the α, β, γ, δ symbols), which has actually no semantic meaning but is useful for the successive translation phase.

The LSs, in fact, are then inputted to the Control Interface which deals with the second stage of translation: in fact, it contains encoders which allow the translation of LSs into Semantic Symbols, SSs, which belong to a Semantic Alphabet (SA) and are meaningful for the Control Interface itself. Some example of SSs are the English Alphabet for a speller application, the items selected into a menu, the directions of the movements of a robotic arm, etc.

SSs are finally mapped into physical controls toward the output application, that can be the movement of a cursor on a screen, a prosthesis, a wheelchair, a speller, and so on.
During the whole duration of a BCI session, some feedback can be provided to the user of the system, that can be a feedback on the classification performances, a feedback on the translation of LSs into SSs or, finally, a feedback from the output peripherals.
As previously mentioned, a lot of different protocols have been implemented in BCI research, according to the features extracted by the brain signals, and different BCI systems have been designed to date, everyone being characterized by different structures and components. However, it has been demonstrated in a recent work (Quitadamo et al., 2008) that a unique standard model is able to describe different BCI systems both from the static point of view and from the dynamic one; in particular, the temporal characterization of different BCI protocols under a unique dynamic scheme is very innovative and meaningful, as it demonstrates that a common basic description can be possible for all them, leading to a standardization of BCI systems.
This is very important for such a complex field as the BCI one, because a unique model avoids that each BCI lab has its own terminology and structures to refer to BCI components, and helps in the process of unification and dissemination of tools and resources among researchers.
The model that has been recently proposed, has been implemented with the Unified Modelling Language (UML) which is a well established visual language for model-driven technologies, which include system development, engineering and architecture. It allows a universal representation of common concepts like classes, components, generalization, aggregation, and behaviors. Also it is very simple to understand, as it makes use of diagrams that even a non-programmer is able to design, modify and reuse. The two most important UML diagrams types that have been used in this model are **class diagrams** and **sequence diagrams.** Class diagrams represent structural correlations among the objects of a system, by means of classes, that are the components of a system, characterized by attributes and methods, that are the operations the class performs. In UML notation classes are represented by means of rectangles divided into compartments, where the name of the class, its attributes and its operations are listed, and connections among classes are represented by means of relationships. Sequence diagrams, instead, describe the temporal relationships occurring between the objects of the system by means of messages running from the sending class to the receiving one.
In this model a clear description of the timing of a trial has been furnished so that the system is characterized not only from the static point of view but from the dynamic one too, that is, the temporal relationships occurring between the objects of the system have been identified. In the following paragraphs some general aspects on the static and dynamic characterizations of the implemented model will be furnished.

3.1 Structure of a trial

The Trial is the main entity of a BCI session, as it leads to the classification of a symbol that can be logical or semantic. For example, in a P300 protocol (Sellers et al., 2006; Krusienski, 2008) a semantic trial leads to the classification of one semantic symbol in a 2D matrix (formed by the English alphabet characters plus the space one and the numbers from 1 to 9); in a μ-rhythm protocol (Wolpaw & McFarland, 2004) a logical trial leads to the selection of a target among those shown on a PC screen. A trial is constituted by a set of processing

quanta (PQs) that are the operations iterated for the update of the internal state of the classifier.
Then four main phases have been identified in a trial, that lead to the classification of a symbol: Preparation, Target Request, Brain Performance and Performance Notification. These four phases are slightly different if the system runs in asking or operating mode; in fact, in asking mode the user of the system is asked to perform a mental task in order to train and test the classifier to recognize his mental states while in operating mode the user communicates what he wants and the system is supposed to be already well-adapted to correctly classify the user intents.
Therefore, in the Preparation phase the system is set to a rest state until the subject is ready to start a BCI experimental session. Then, in the Target Request phase, only possible if the system runs in asking mode, the subject is requested to perform a mental task.
In the Brain Performance phase the subject generates a symbol as a consequence of a specific mental task; his mental states are then classified in order to generate a LS or a SS and he can be provided with a feedback for the adjustments on the performances. The result of the classification is finally communicated to the subject in the Performance Notification phase. At the end of the last phase a rest period can occur.
In Fig. 2 a schematic representation of a trial in a μ-rhythm protocol is reported: in asking mode, after the warning message at the end of the preparation phase, the subject is notified with the request of the mental task. In this example he has to select the middle target among those represented on the monitor; the reaching of the desired target on the screen corresponds to the generation of a LS. In the Brain Performance phase he does his mental task to reach the target and, at the end of each PQ, the classifier updates its internal state. A symbol is generated at the end of this phase and the result is notified to the subject in the Performance Notification. In operating mode, during which there is not a Target Request phase, after the warning, the subject performs mental tasks to select the bottom target among those available.

3.2 Actors of the system

Four main actors have been defined which attend to the correct execution of a trial and that in the UML notation are identified by classes: the Session Manager, the Transducer, the Control Interface and the Feedback Manager.
The Session Manager is active for the whole duration of the BCI experimental session and, in particular, it manages the trial execution by invoking methods on all the other actors.
The Transducer is constituted by the collector and the classifier (Features Extractor and Features Translator) so it acquires the physiological signals and attends to their translation into LSs.
The Control Interface encodes sequences of LSs into SSs and stores the classification performances history. It would be noteworthy to avoid direct dependence relationship between the transducer and the control interface in order to realize a better software modularization.
The Feedback Manager handles both the feedback in the Brain Performance phase (for example, after getting the results of the analysis performed by the Transducer on the physiological signals, it updates the position of a cursor on the screen) and the feedback to the system's user, by notifying special signals and events, and in general manages all the features regarding the direct interface with the subject.

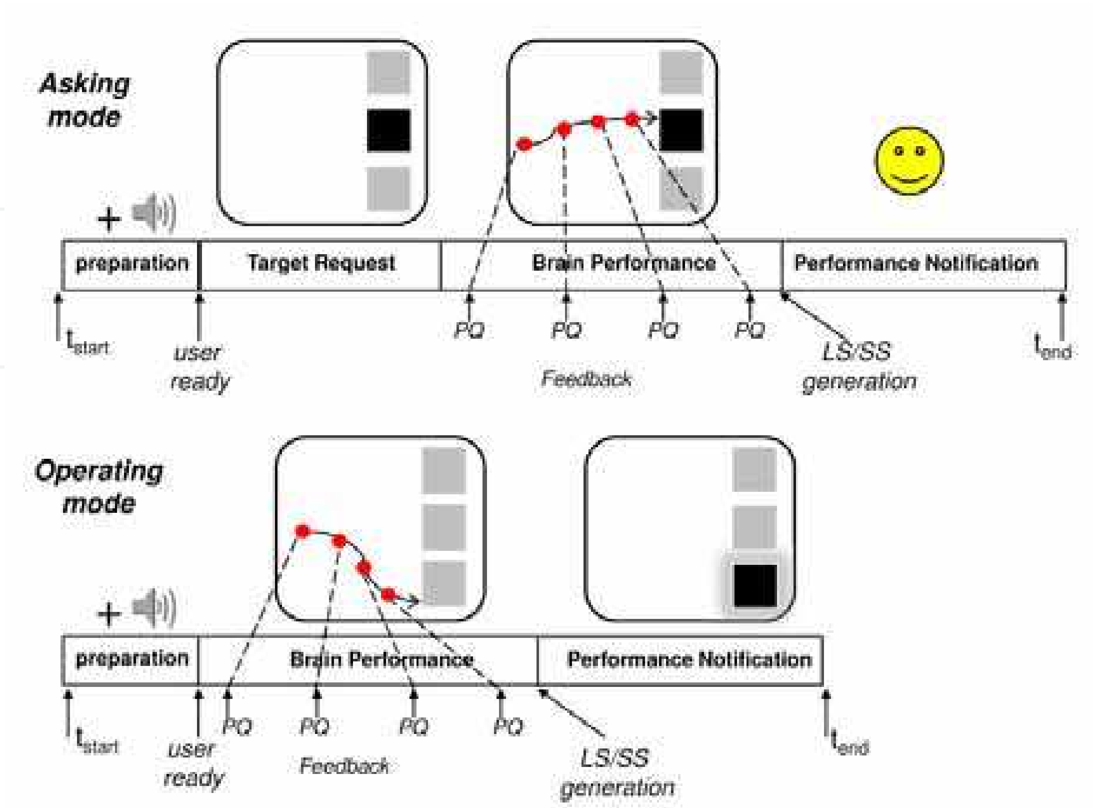

Fig. 2. Structure of a trial

In Fig. 3 a class diagram of the four actors is reported. Dotted arrows denote dependency relationships between actors; black diamonds are for composition relationships, as for the transducer class which is composed by the classifier and the collector module.

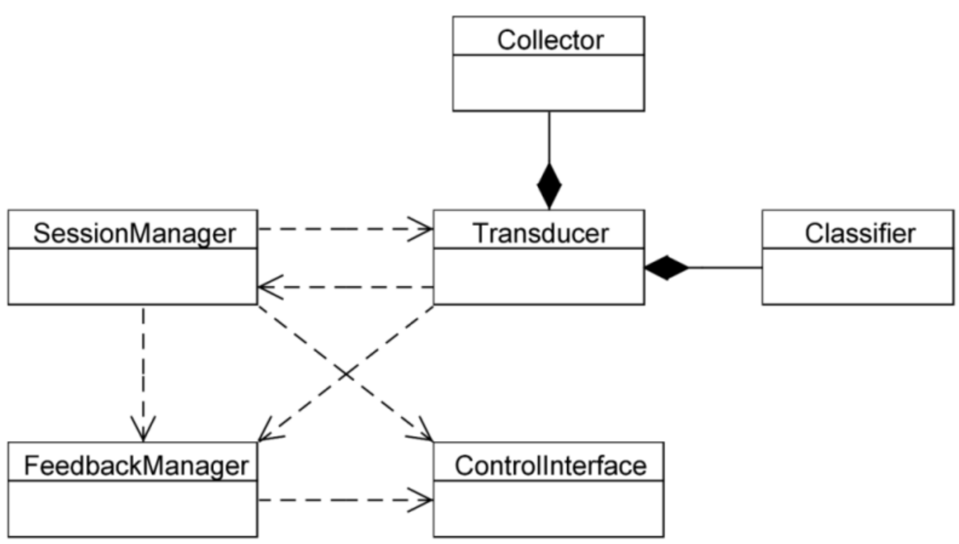

Fig. 3. Class diagram of the actors

3.3 Timing of a trial

Up to now static features of the model have been described. However a static characterization of BCI systems is not sufficient if, for example, an easy substitution of hardware and software modules is desirable; in fact, the temporal sequence of operations tells when and in which manner two different objects of the system communicate each other and if this communication is not standardized, how can different modules be interchanged?
For this reasons, the timing of a trial is the most important novelty that the model proposed in (Quitadamo et al., 2008) can add to those already presented in the literature: the identification of the flux of operations that characterize a typical BCI trial is fundamental for the demonstration that a unique model is able to describe different protocols.
Also, by means of this model, it is possible to define unique interfaces that allow to freely substitute a module with another one: Lab 1 could use Lab 2's classifier without any effort, or an EEG manufacture could simply provide a driver so that the system can freely be changed according to the specific requirements.
Note that here it will be illustrated a simplified version of the whole model that has been fully described in (Quitadamo et al., 2008): so the reader can refer to the cited article for a more detailed description of the model and for having an idea on how it can be easily used to explain five BCI protocols (P300, SCP, μ-rhythms, SSVEP, fMRI).
The timing of the trial has been UML-modeled by means of a sequence diagram (Fig. 4) in which the four actors previously mentioned communicate by means of messages. From the diagram it is possible to follow the temporal evolution of the different phases of the trial and the way each actor acts during them.
The Session Manager resets the internal clock of the system after receiving a message from the outside which invokes the starting of a trial (*StartTrial()*). In asking mode (*opt,* optional operations), the Session Manager notifies the asked symbol (the logical mental task to perform, the SS to select) to the Control Interface, which manages the encoding strategies, and to the Feedback Manager (*setAskedSymbol()*), which is responsible of the communication of the symbol to the subject. Then it communicates the starting of the trial to all the other actors which eventually get ready (*onTrialBegin()*). The Feedback Manager notifies to the subject that the active phase is going to start (for example by means of a sound, a fixation cross, etc.) and which task he is asked to perform (*onRequestNotify()*). Then the processing phase begins (*onProcessingBegin()*), with a loop of operations in which the Session Manager scans the starts of the PQs to the Transducer (*onPQBegin()*), which on its turn communicates the starts and the ends of the PQs to the Feedback Manager (*onPQBegin()*, *onPQEnd()*); the Feedback Manager gets the analysis results (*getAnalysisResult()*) and updates the feedback if necessary (for example, it updates the position of a cursor on a screen).
When the classification of a symbol is reached, that is the Classifier is able to give a result (*alt* fragment indicates an alternative), or the time established for it has elapsed (the internal clock of the system interrupts the classification and the Classifier abstains from giving a result), the trial ends (*EndTrial()*); the Transducer communicates the classification result to the Session Manager which communicates it to all the remaining actors (*onClassificationReached()*). Finally the Feedback Manager notifies the overall performance to the subject (*onPerformanceNotify()*) and the trial definitively ends.
It is important to underline that all the protocols previously mentioned have successfully been described with this unique temporal structure and this can be considered a preliminary step towards the unification of BCI systems.

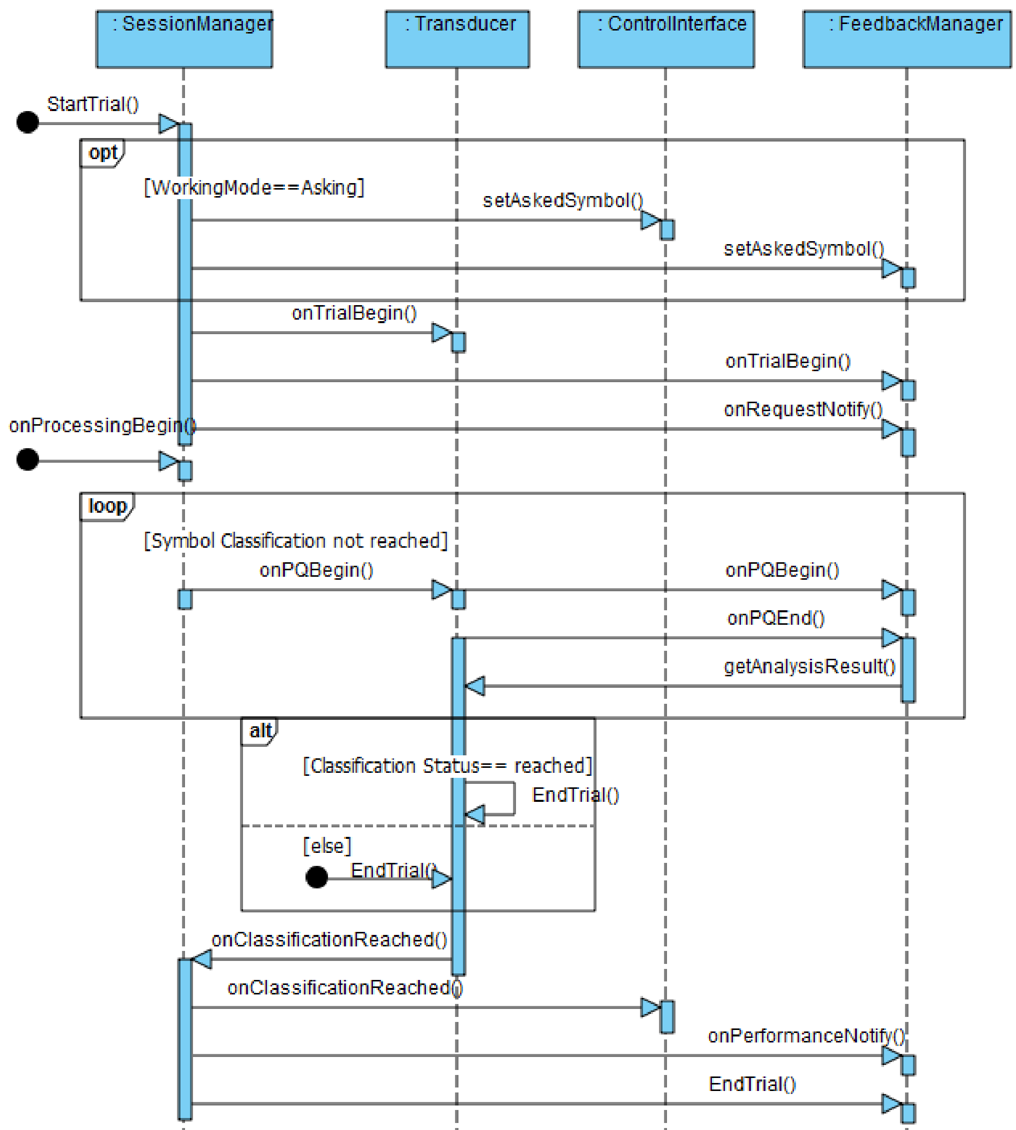

Fig. 4. Timing of a trial

4. A new performance indicator: the Efficiency

The evaluation of the performances of BCI systems is a capital item in BCI research. Different metrics have been proposed to date, such as Bit-Rate, Error-Rate and Mutual Information (Schlögl et al., 2003; Wolpaw, 2002), but all of them are somehow incomplete in characterizing a real-world application, as they do not take into account, for example, errors consequences and strategies to correct them. In particular, errors consequences strictly depend upon the final application that the BCI is intended to pilot: if the subject is communicating by means of a P300 speller, an error into communication is something that involves simply the selection of an UNDO command and a new mental task to reselect the desired character; in these cases errors do not affect severely system performances. However, if the BCI is piloting a wheelchair, an error in the detection of the exact intents of the subject can lead to situations that could be dangerous for the subject himself. These last

are particular cases in which errors weight a lot in the evaluation of the system performances and it is preferable that the classifier abstains from classification unless the certainty on it is maximal.
For all these reasons, a new metric has been proposed which evaluates the performances of a system as a function of the performances of both the Transducer and the Control Interface (Bianchi et al., 2007). This metric is strictly correlated to the functional model described in the previous section and gives some costs to the errors in the communication, so that a more reliable evaluation of systems is achievable.

4.1 Describing the Transducer performances: the Extended Confusion Matrix

The Extended Confusion Matrix (ECM) evaluates the performances of a transducer, that is, it stores the results of the classifications of the mental tasks performed by the subject. In facts it represents on the rows the LSs to be classified and on the columns the LSs actually classified; the last column of the matrix represents the cases in which the classifier was not able to classify and so abstained from decisions (*abstaining classifiers*). Abstentions are really important in such kinds of application which must be as much safe as possible. Then the elements of the principal diagonal are the correct classifications, while the remaining non-diagonal elements constitute the wrong ones.
As an example, suppose that the mental tasks that the subject is required to perform are associated to a logical alphabet of four logical symbols, α, β, γ, δ.
An example of ECM is now reported:

$$ECM = \begin{array}{c|c|c|c|c|c} & \alpha & \beta & \gamma & \delta & abst \\ \hline \alpha & 84 & 8 & 0 & 8 & 0 \\ \hline \beta & 4 & 96 & 0 & 0 & 0 \\ \hline \gamma & 12 & 0 & 80 & 8 & 0 \\ \hline \delta & 12 & 4 & 12 & 72 & 0 \end{array} \quad (1)$$

As it can be seen from (1), most of the classification errors are associated to the generation of the fourth symbol δ; this is of great importance in the choice of the encoding strategy to implement, as selecting an encoding which minimizes the occurrence of the δ symbol will result in the minimization of the errors and this can represent an optimization strategy.
From an ECM, a Misclassification Probability Matrix (MPM) can be defined in the following way (N_{LA} is the length of the logical alphabet) (eq. 2-5):

$$MPM[i,j] = \frac{ECM[i,j]}{n_i}, i \neq j \quad (2)$$

$$MPM[i,j] = 0, i = j \quad (3)$$

$$n_i = \sum_{j=1}^{N_{LA}+1} ECM[i,j] \tag{4}$$

$MPM =$ (5)

	α	β	γ	δ	$abst$
α	0	0.08	0	0.08	0
β	0.04	0	0	0	0
γ	0.12	0	0	0.08	0
δ	0.12	0.04	0.12	0	0

and an Extended Overtime Matrix (EOM) is defined as in (6):

$EOM =$ (6)

	α	β	γ	δ	$abst$
α	0	2	2	2	1
β	2	0	2	2	1
γ	2	2	0	2	1
δ	2	2	2	0	1

The MPM represents the probabilities concerning incorrect classifications and EOM represents the "costs" associated to errors and indeterminateness cases and which depend on the Control Interface error correction strategy. In the example in (6), a cost of "0" means that no error was done during the selection of the LS; a cost of "1" is associated to indeterminate cases and means that, when a classifier abstains from a decision, only a reselection of the correct LS is needed; finally a cost of "2" means that, after an error, one has to perform two additional steps to correct the error, that is, one to delete the wrong LS and one to reselect the right one. However, different correction strategies can be implemented, which can be somehow smart; for example they can take into account some thresholds for the decision of the correct symbol to be classified so reducing the cost associated to some errors. The strategies therefore are characteristic of the Control Interface and can be manifold according to the requirements of the system itself.

4.2 Measuring error consequences: Super Tax Vector

When a misclassification or an abstention occur the loss of information can be quantified with the Super Tax Vector, whose i-*th* element is given by:

$$ST(i) = \sum_{j=1}^{N_{LA}+1} MPM[i,j] \cdot EOM[i,j] \tag{7}$$

$$ST = \begin{bmatrix} 0.32 \\ 0.08 \\ 0.40 \\ 0.56 \end{bmatrix} \tag{8}$$

where each element of ST represents the fraction of additional selections that are necessary to correct a mistake. In (8), the (7) has been applied to our example.

One can easily see that the generation of the fourth symbol appears to be the most difficult, as already pointed out. In fact, the ST value associated to it is the highest one.

The elements of the ST vector (ST(i)) represent the additional selections that must be done to recover an error; however, these additional selections are not error-free, as they imply a classification too, and the classification process implies misclassifications on its turn. Misclassifications lead to the same ST(i) elements, so that $ST(i)^2$ new selections are needed, that are on their turn subject to misclassification.

The geometric series can be recognized in (9):

$$\sum_k ST(i)^k = 1 + ST(i) + ST(i)^2 + ST(i)^3 + \ldots \tag{9}$$

which converges to (10):

$$\sum_k ST(i)^k = \frac{1}{1 - ST(i)}, \text{ if } |ST(i)| < 1 \tag{10}$$

Finally, the Expected mean Selection Cost (ESC), that is the number of classifications required to generate a correct LS, is defined as in (11):

$$\overline{ESC} = \sum_{j=1}^{N_{LA}} \frac{p_{occ}(i)}{1 - ST(i)} \tag{11}$$

where $p_{occ}(i)$ is the probability of the i^{th} LS to occur.

If N_{SA} is the length of the SA, $p_{SS}(n)$ is the probability of the n^{th} SS to occur and $l(n)$ is the number of LSs used for the encoding, then the mean codeword length can be defined as in (12):

$$\overline{L_{CW}} = \sum_{j=1}^{N_{LA}} p_{ss}(n) \cdot l(n) \tag{12}$$

An example of an encoding map between LA and SA (the English Alphabet plus the space character) is given in Table 1. Note that the symbol δ is used as an UNDO key. Choosing different encodings (e.g. variable length encodings), L_{CW} can be reduced, leading to a compression of the message.

CHAR	E	CHAR	E	CHAR	E
A	ααα	J	βαα	S	γαα
B	ααβ	K	βαβ	T	γαβ
C	ααγ	L	βαγ	U	γαγ
D	αβα	M	ββα	V	γβα
E	αββ	N	βββ	W	γββ
F	αβγ	O	ββγ	X	γβγ
G	αγα	P	βγα	Y	γγα
H	αγβ	Q	βγβ	Z	γγβ
I	αγγ	R	βγγ	Sp	γγγ

Table 1. Example of an encoding map between SA and LA.

Finally the Efficiency (*Eff*) of a system is defined as in (13):

$$Eff = \frac{1}{\overline{L_{CW}} \cdot \overline{ESC}} \tag{13}$$

It is easy to see that the *Eff* of a system depends on both the Transducer (L_{CW}, ECM and MPM) and the Control Interface (EOM) even if they have been separately implemented; so the choice of the best BCI system to adapt to the user requirements can be performed only after the evaluation of the best Transducer-Control Interface combination.
It has been demonstrated in (Bianchi et al., 2007) that Efficiency is a better indicator of the performances of a BCI system than Mutual Information in a copy-spelling task.

5. Optimizing BCI systems: BF++Toys

The metric and the file format defined in the previous paragraphs allowed the implementation of a set of tools dedicated to the simulation of BCI systems and to the prediction of the their performances (Quitadamo, 2007). They were called BF++Toys because they are included in the Body Language Framework (Bianchi et al., 2003) and have been developed in C++. Their main feature is the open-source nature that makes it possible for everyone to modify and adjust them according to the need of his system. They can be downloaded at www.brainterface.com.
BF++ Toys deal with logical and semantic alphabets, encoders, confusion matrices, errors correction strategies and are actually formed by four main applications:

- the ECM generator, which can simulate different transducers with different errors and abstentions distributions and finally create new ECMs;
- the Encoding Generator, which can create new encoders by using a LA and a SA;
- the Simulator, which, given an ECM and an encoding, can generate realistic sequences of LSs in copy spelling task;

- the Optimizers, which can find the best combination of Transducers and Control Interfaces on the basis of the metric described in the previous paragraphs.

The most important thing that has to be underlined here, is that BF++ Toys were a direct derivation of the XML file format and the BCI definitions previously illustrated; this means that they can be used to optimize whatever BCI system independently from the protocol to be implemented (P300, μ-rhythms, etc.) or from the final application, as they entirely respect the described standards.
Finally, the possibility to simulate different BCI systems and to find the solution with the highest efficiency is very valuable as it can help in the choice of the system that best fits to the users' requirements.

6. Conclusions

The necessity of standards in the research is effective as they can enhance the dissemination of data and information among all the groups that are interested in it or want to cooperate. The main goal of standards is to harmonize the resources available in a particular research field and create a common language for defining its components, that anyone can refer to, by avoiding confusion or misunderstandings due to different semantics.
This necessity is compelling mostly in the BCI research field, where a lot of systems, protocols, frameworks, file formats, have been established to date, lacking a concrete collaboration among different groups and so leading to a concrete difficulty when data have to be exchanged, different systems have to be evaluated and compared, different hardware and software modules have to be substituted, etc.
Standards in BCI can be defined on three different main levels: file formats, functional model and methods. These three levels are strictly interconnected and functional to each other: in fact, while a common file format is essential for an easy storage and exchange of all the data related to BCI, a functional model with standard definitions allow to use a common language to talk about BCI and so makes the dissemination of BCI-related resources much easier. Finally, common methods for evaluating the performances of systems allow the comparison of different systems and so the choice of the most suitable system for the final application.
In conclusion all the three levels together lead to common shared tools whose ultimate aim is to build systems that definitively fit to the patients' residual skills and requirements and from which they can derive real benefits to improve the quality of their lives.

7. Acknowledgments

This work is partially supported by the European ICT Programme Project FP7-224631 and by the DCMC Project of the Italian Space Agency. This paper only reflects the authors' views and funding agencies are not liable for any use that may be made of the information contained herein.

8. References

Bianchi, L.; Babiloni, F.; Cincotti, F.; Salinari, S. & Marciani, M. G. (2003). Introducing BF++: A C++ Framework for Cognitive Bio-Feedback Systems Design. *Methods of Information in Medicine,* Vol. 42, No. 1, (July 2003) 104-110, 0026-1270.

Bianchi, L.; Quitadamo, L. R.; Garreffa, G.; Cardarilli, G. C. & Marciani, M. G. (2007a). Performances evaluation and optimization of brain computer interface systems in a copy spelling task. *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* Vol. 15, No. 2, (June 2007) 207-16, 1534-4320.

Bianchi, L.; Quitadamo, L. R.; Marciani, M. G.; Maraviglia, B.; Abbafati, M. & Garreffa, G. (2007b). How the NPX data format handles EEG data acquired simultaneously with fMRI. *Magnetic Resonance Imaging,* Vol. 25, No. 6, (July 2007) 1011-4, 0730-725X.

Cincotti, F.; Bianchi, L.; Birch, G.; Guger, C.; Mellinger, J.; Scherer, R.; Schmidt, R. N.; Yáñez Suárez, O. & Schalk, G. (2006). BCI meeting 2005--workshop on technology: hardware and software. *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* Vol. 14, No. 2 (June 2006) 128-31, 1534-4320.

Coyle, S. M.; Ward, T. E. & Markham, C. M. (2007). Brain-computer interface using a simplified functional near-infrared spectroscopy system. *Journal of Neural Engineering,* Vol. 4, No. 3, (September 2007) 219-26, 1741-2560.

Donchin, E.; Spencer, K. M. & Wijesinghe, R. (2000). The Mental Prosthesis: Assessing the Speed of a P300-Based Brain-Computer Interface. *IEEE Transactions on Rehabilitation Engineering,* Vol. 8, No. 2, (June 2000) 174-179, 1063-6528.

Kemp, B. & Olivan, J. (2003). European data format 'plus' (EDF+), an EDF alike standard format for the exchange of physiological data. *Clinical Neurophysiology,* Vol. 114, No. 9, (September 2003) 1755-1761, 1388-2457.

Krusienski, D. J.; Sellers, E. W.; McFarland, D. J.; Vaughan, T. M. & Wolpaw, J. R. (2008). Toward enhanced P300 speller performance. *Journal of Neuroscience Methods,* Vol. 167, No. 1, (January 2008) 15-21, 0165-0270.

Kübler, A. & Neumann, N. (2005). Brain-computer interfaces-the key for the conscious brain locked into a paralyzed body. *Progress in Brain Research,* Vol. 150, (2005) 513-25, 0079-6123.

Mason, S. G. & Birch, G. E. (2003). A general framework for brain-computer interface design. *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* Vol. 11, No. 1, (March 2003) 70-85, 1534-4320.

Mellinger, J.; Schalk, G.; Braun, C.; Preissl, H.; Rosenstiel, W.; Birbaumer, N. & Kübler, A. (2007). An MEG-based brain-computer interface (BCI). *NeuroImage,* Vol. 36, No. 3, (July 2007) 581-93, 1053-8119.

Quitadamo, L. R.; Marciani, M. G.; Cardarilli, G. C. & Bianchi, L. (2008). Describing different brain computer interface systems through a unique model: a UML implementation. *Neuroinformatics,* Vol. 6, No. 2, (June 2008) 81-96, 1559-0089.

Quitadamo, L. R; Marciani, M. G & Bianchi, L. (2007). Optimization of Brain Computer Interface systems by means of XML and BF++ Toys. *International Journal of Bioelectromagnetism,* Vol. 9, No. 3, (December 2007) 172-184.

Schalk, G.; McFarland, D. J.; Hinterberger, T.; Birbaumer, N. and Wolpaw, J. R. (2000) BCI2000: a general-purpose brain-computer interface (BCI) system. *IEEE Transactions on Biomedical Engineering*, Vol. 51, No. 6, (June 2000) 1034-43, 0018-9294.

Schalk, G.; Miller, K. J.; Anderson, N. R.; Wilson, J. A.; Smyth, M. D.; Ojemann, J. G.; Moran, D. W.; Wolpaw, J. R. & Leuthardt, E. C. (2008). Two-dimensional movement control using electrocorticographic signals in humans. *Journal of Neural Engineering,* Vol. 5, No. 1, (March 2008) 75-84, 1741-2560.

Schlögl, A. (2006) GDF - A general dataformat for BIOSIGNALS. *cs/0608052* (Aug 2006), available at http://arxiv.org/abs/cs/0608052.

Schlögl, A.; Keinrath, C.; Scherer, R. & Pfurtscheller G. (2003). Information transfer of an EEG-based brain computer interface, *Proceedings of the 1st International IEEE EMBS Conference on Neural Engineering*, pp. 641-644, Capri Island, Italy, March 20-22, 2003.

Sellers, E. W.; Krusienski, D. J.; McFarland, D. J.; Vaughan, T. M. & Wolpaw, J. R. (2006). A P300 event-related potential brain-computer interface (BCI): the effects of matrix size and inter stimulus interval on performance. *Biological Psychology,* Vol. 73, No. 3, (October 2006) 242-52, 0301-0511.

Weiskopf, N.; Mathiak, K.; Bock, S. W.; Scharnowski, F.; Veit, R.; Grodd, W.; Goebel, R. & Birbaumer, N. (2004). Principles of a brain-computer interface (BCI) based on real-time functional magnetic resonance imaging (fMRI). *IEEE Transactions on Bio-Medical Engineering,* Vol. 51, No. 6, (June 2004) 966-70, 0018-9294.

Wolpaw, J. R. & McFarland, D. (2004). Control of a two-dimensional movement signal by a non invasive brain-computer interface in humans. *Proc. Natl. Acad. Sci,* Vol. 101, No. 51, (December 2004) 17849-17854, 0027-8424.

Wolpaw, J. R.; Birbaumer, N.; McFarland, D. J.; Pfurtscheller, G. & Vaughan, T. M. (2002). Brain-computer interfaces for communication and control. *Clinical Neurophysiology,* Vol. 113, No. 6, (June 2002) 767-91, 1388-2457.

Yoo, S. S; Fairneny, T.; Chen, N. K; Choo, S. E.; Panych, L. P.; Park, H.; Lee, S. Y. & Jolesz, F. A. (2004). Brain-computer interface using fMRI: spatial navigation by thoughts. *Neuroreport,* Vol. 15, No. 10, (July 2004) 1591-5, 0959-4965.

Biomedical Engineering

Edited by Carlos Alexandre Barros de Mello

ISBN 978-953-307-013-1
Hard cover, 658 pages
Publisher InTech
Published online 01, October, 2009
Published in print edition October, 2009

Biomedical Engineering can be seen as a mix of Medicine, Engineering and Science. In fact, this is a natural connection, as the most complicated engineering masterpiece is the human body. And it is exactly to help our “body machine” that Biomedical Engineering has its niche. This book brings the state-of-the-art of some of the most important current research related to Biomedical Engineering. I am very honored to be editing such a valuable book, which has contributions of a selected group of researchers describing the best of their work. Through its 36 chapters, the reader will have access to works related to ECG, image processing, sensors, artificial intelligence, and several other exciting fields.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821