Journal of Neuroscience Methods 175 (2008) 154–162

Contents lists available at ScienceDirect

Journal of Neuroscience Methods

journal homepage: www.elsevier.com/locate/jneumeth

A new concept of a unified parameter management, experiment control, and data analysis in fMRI: Application to real-time fMRI at 3 T and 7 T

M. Hollmann[a,*], T. Mönch[a], S. Mulla-Osman[a], C. Tempelmann[b], J. Stadler[c], J. Bernarding[a]

[a] Department of Medical Informatics, Otto-von-Guericke University, Leipziger Straße 44, 39120 Magdeburg, Germany
[b] Clinic for Neurology II, Otto-von-Guericke University, Magdeburg, Germany
[c] Leibniz Institute for Neurobiology, Magdeburg, Germany

ARTICLE INFO

Article history:
Received 25 February 2008
Received in revised form 7 August 2008
Accepted 12 August 2008

Keywords:
Real-time fMRI
Brain computer interface
Adaptive stimulus control
Intelligent imaging
7 T
High field MRI

ABSTRACT

In functional MRI (fMRI) complex experiments and applications require increasingly complex parameter handling as the experimental setup usually consists of separated soft- and hardware systems. Advanced real-time applications such as neurofeedback-based training or brain computer interfaces (BCIs) may even require adaptive changes of the paradigms and experimental setup during the measurement. This would be facilitated by an automated management of the overall workflow and a control of the communication between all experimental components. We realized a concept based on an XML software framework called *Experiment Description Language* (EDL). All parameters relevant for real-time data acquisition, real-time fMRI (rtfMRI) statistical data analysis, stimulus presentation, and activation processing are stored in one central EDL file, and processed during the experiment. A usability study comparing the central EDL parameter management with traditional approaches showed an improvement of the complete experimental handling. Based on this concept, a feasibility study realizing a dynamic rtfMRI-based brain computer interface showed that the developed system in combination with EDL was able to reliably detect and evaluate activation patterns in real-time. The implementation of a centrally controlled communication between the subsystems involved in the rtfMRI experiments reduced potential inconsistencies, and will open new applications for adaptive BCIs.

1. Introduction

Neuroimaging techniques often use quite complex experimental setups. As one of the most widespread neuroimaging techniques functional magnetic resonance imaging (fMRI) studies usually require several hard- and software components. These components handle the data acquisition, stimulus presentation, and the data analysis. The different components and relevant procedures require consistency checks for parameter interdependencies within the different subsystems such as, e.g. the number of presented stimuli and their presentation duration with respect to the repetition time. Usually these checks are performed manually by the investigator. A central and automated control of all parameters would not only ease the handling of the experiment, but also significantly lower potential errors due to inconsistent data analyses. It would additionally render the implementation of real-time fMRI (rtfMRI) applications, such as biofeedback experiments (Weiskopf et al., 2004a; DeCharms et al., 2004, 2005) or brain computer interfaces (BCIs) (Yoo et al., 2004) much more user friendly and less prone to errors.

In a typical neurofeedback experiment the subject tries to lower or increase the activation of a circumscribed brain area, thereby altering his/her own mental state (Posse et al., 2003, 2007; DeCharms et al., 2005). Extending this concept to a training situation, duration and content of a stimulus presentation may depend on the subject's performance. This, in turn, would require that the whole experimental setup including stimulus presentation, communication streams, and real-time data analysis has to be dynamically adapted in a consistent way during the experiment.

Additionally a simulation of an experiment prior to the execution would be extremely helpful to detect potential errors or misinterpretations already at an early stage. So far, a few approaches exist to unify complex fMRI/rtfMRI experiments (Smyser et al., 2001; Voyvodic, 1999; Eloquence, 2006; Weiskopf et al., 2004b). However, these approaches have in common that no meta information is used to connect involved systems, but the unification of the different hard- and software components is realized within the closed software code. Therefore, the extension or

* Corresponding author at: Tel.: +49 391 67 13535; fax: +49 391 67 13536.
E-mail address: Maurice.Hollmann@med.ovgu.de (M. Hollmann).

0165-0270/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.jneumeth.2008.08.013

adaptation of these software systems to new experimental conditions requires specialized solutions. On the contrary we aimed to build a framework where the parameters of the experimental setup are hold in a central description, which allows performing cross-validation checks and the usage of the stored information in any application by implementing the according interfaces. The details of differences between our concept and the above-mentioned approaches will be outlined in Section 4.

Analyzing the general workflow of a typical rtfMRI experiment shows that at least four subsystems are connected to form the standard setup: (1) the *data acquisition* subsystem that includes the scanner software for parameter handling and data acquisition control as well as the image reconstruction, image storage, and image export; (2) the *stimulus presentation* subsystem including a bidirectional communication between the scanner and the stimulus presentation unit. Here, some sort of information exchange, e.g. sending trigger signals to start the scanning process, or receiving trigger events to start the stimulus presentation have to be included; (3) the *data preprocessing* and *statistical data analysis* that serves to determine activated areas; and (4) an *activation processing subsystem* that is used for further processing, classification, and interpretation of the extracted activation patterns. Additional systems or interfaces, such as to simultaneously record physiological data, may be implemented.

Although these disconnected systems and their parameter sets are highly interdependent each subsystem is usually represented by its own software, often run on a separate computer, and controlled by specific parameters in different formats. Two major difficulties arise from this system architecture. First, the error-proneness increases with an increasing number of parameters as well as with increasing interdependencies. A typical example is a multi-stage experiment where the results of initial measurements are used to adapt the parameter settings of the subsequent experiments. The interaction between the subsystems, which is also a prerequisite in experiments with dynamic paradigms and BCIs, is also difficult to implement without some sort of a central parameter representation and the according timing of the communication between the subsystems. Additionally, cross-checks of the parameters have to be included to validate the interaction between all parameters of the subsystems. For very advanced experimental setups, such as dynamic inter-experimental parameter changes due to activation-dependent setups, the communication between the subsystems must be consistent and well-defined.

Based on this analysis we developed a novel concept to realize a unified central control of the complete fMRI experiment including data post-processing and data analysis. The implemented structure consists of different modules that can be combined for the particular experiments. The flexible parameter management can also be used for a real-time adaptation of the data analysis and the stimulus presentation (dynamic experiments).

The presented solution fulfills the above requirements by employing a uniform parameter management with an integrated flexible parameter description that serves to control all different subsystems in a consistent manner. Based on XML (Extensible Markup Language) (XML, 1996–2007) an *Experiment Description Language* (EDL) was developed. EDL is used to control, and thereby to connect, four separated systems (*MRI data acquisition*, *stimulus presentation*, (*real-time*) *statistical analysis of fMRI data*, and *activation processing*). See Fig. 1 for an overview of the central parameter management. An EDL file can be edited using a custom-made application called EDL editor.

The system is designed modularly, thus separated software parts (C++ or Matlab modules) can be replaced or the system may be extended by additional modules. The modules and the workflow of our experiments will be described in further detail within the methods part. Subsequently, a proof-of-principle experiment was performed to validate the concept for practical rtfMRI applications.

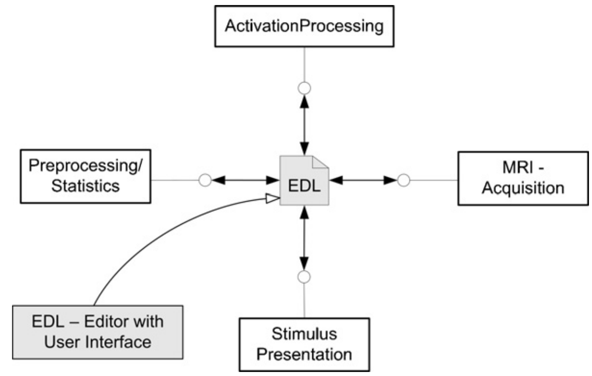

Fig. 1. Scheme of the information infrastructure. EDL is the central information repository where all information is stored to control the fMRI experiment. Each subsystem is connected to the EDL file via a bidirectional interface represented by the circle. This technique ensures highest flexibility of the incorporated software components.

2. Materials and methods

2.1. Technical infrastructure

System implementation and test experiments were conducted at two whole body MRI scanners (3 T Trio, and 7 T equipped with Avanto Gradients, both Siemens Medical Systems, Erlangen, Germany) at the University of Magdeburg, Clinic for Neurology II, and the Leibniz Institute for Neurobiology, Magdeburg. An eight element phased array was used for imaging at both scanners.

On each site, the vendor's EPI BOLD sequences (version VA25A at 3 T and VB12H at 7 T) and the corresponding reconstruction programs were modified to export each volume dataset immediately after acquisition and internal motion correction in real-time to the host computer of the MR scanner (see Fig. 2 for a scheme of the hardware and the data flow). For this purpose a functor (a functional module in a Siemens MR sequence) was added to the standard functor chain of the image reconstruction component of the EPI sequence. This functor performs the image export to any specified host in the local area network.

The statistical data analysis was performed on an external computer ("*Statistics PC*", Pentium IV, 3.0 GHz, 2 GB Random Access Memory (RAM), Windows XP), connected to the host via a 100 MBit/s network. The custom-made software running on this system for statistical data analysis and activation processing (classification) was implemented in Matlab (version R2006b, Matlab, 1984–2006).

A second PC ("*Stimulus PC*", Pentium IV, 2.4 GHz, Windows XP, 2 GB RAM) served for the presentation of visual and auditory stimuli. Visual information was projected with a video projector on a transparent screen and viewed via a 45° mirror mounted on the receiver coil.

The presentation computer was linked to the *Statistics PC* via a 100 MBit/s network connection. This architecture (Fig. 2) ensured a direct communication as prerequisite for dynamic adaptations of the data evaluation mode and the presentation of stimuli. All components are independent of the specific scanner, except for the interfaces for the real-time export of the image data that had to be implemented for the vendor-specific scanner system.

An EDL file forms the central repository for parameters of all connected systems. The *statistical analysis* and *activation processing* software imports required parameters from the central EDL file that

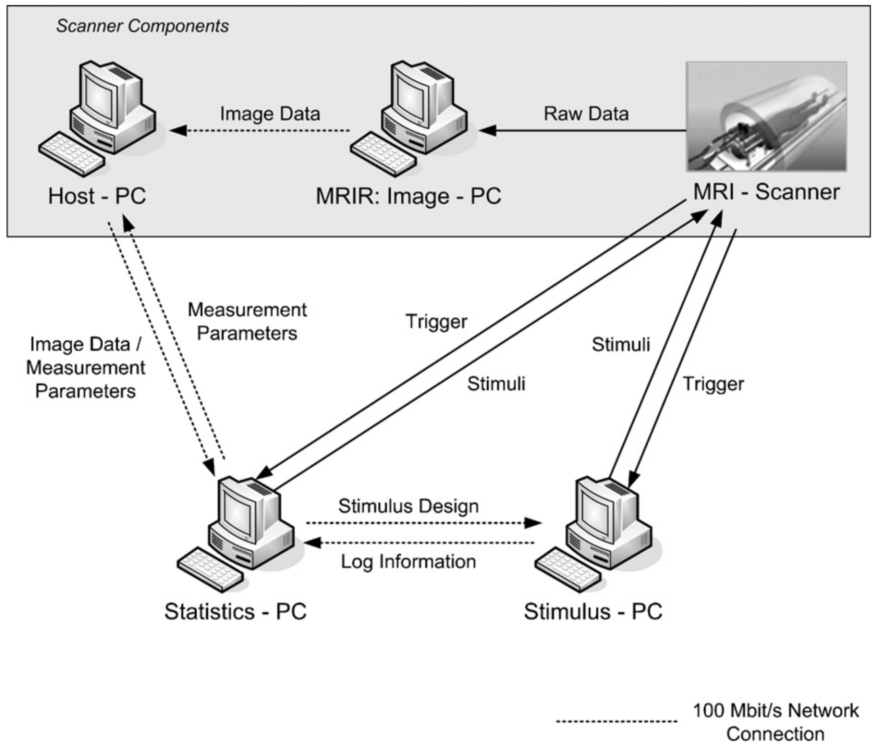

Fig. 2. Hardware scheme for the real-time fMRI setup. The components highlighted in gray depict the vendor-specific measurement system. During an experiment the original MR data are Fourier-transformed and motion-corrected by the vendor image processing unit (*Image PC*, MRIR). The reconstructed data are transferred in real-time to the host computer where the statistical analysis (at the *Statistics PC*) of the data is performed using custom-made software. Stimuli can either be presented by the *Statistics PC* or for more demanding applications (such as virtual reality environments) by the *Stimulus PC*. In the latter case the stimulus presentation is controlled by the EDL representation at the *Statistics PC*. The trigger connects both the *Statistics PC* and *Stimulus PC*, and serves for the synchronization of the system.

is stored on the *Statistics PC*. While measurement parameters can be transferred between EDL and the *Host PC* the stimulus presentation at the *Stimulus PC* is also EDL-driven.

It is important to note that any actually executed experiment is defined in an EDL file just as any related initial experiment. In this case an EDL file can contain references to the EDL files of prior measurements. This technique is used to connect several parts of an experiment, e.g. a neurofeedback measurement and the according functional localizer measurements.

2.2. EDL—Experiment Description Language

The essence of our concept is a central information repository (Fig. 1) that serves as a control and validation node for each of the separated modules. As the repository contains the complete parameter set of all experimental parameters, the semantic relations of the experiment such as interaction and design aspects of the different parameter sets can be checked automatically.

The experimental parameters of each subsystem are stored within this repository in terms of a newly developed XML-based *Experimental Description Language* (EDL). XML is a widespread standard meta-language used to define other languages using structural definitions stored in *XSD* (*XML schema definition*). Therefore, EDL documents can be validated using predefined structural information which contains the internal hierarchy as well as parameter limits and default values of the defined elements.

For the validation of the structural correctness, public domain standard XML libraries can be used which eases the information handling. We used the free Xerces-C++ (Xerces-C++, 2001–2005) library in version 2.70. The parameters within the central file can be modified interactively either by the user or by the system itself, thereby allowing an adaptive but parameter-coherent system even during run time.

An EDL file consists of several components containing the required information in a typical XML hierarchical structure. In the current version (EDL 1.2) seven sections form the basic structure of an experiment (see Fig. 3).

The following small EDL code fragment exemplarily shows the underlying structure for an element that specifies which preprocessing and statistical analysis will be used in the data analysis part of this specific experiment:

```
<CorrAnalysis windowType="constant">
  <spatialFilter>
    <filterSize>3</filterSize>
  </spatialFilter>
  <CGamma>
    <tPeak1>6</tPeak1>
    <mWidth1>5.1</mWidth1>
    <scale1>1</scale1>
    <tPeak2>16</tPeak2>
    <mWidth2>8</mWidth2>
    <scale2>0.09</scale2>
    <offset>0</offset>
  </CGamma>
</CorrAnalysis>
```

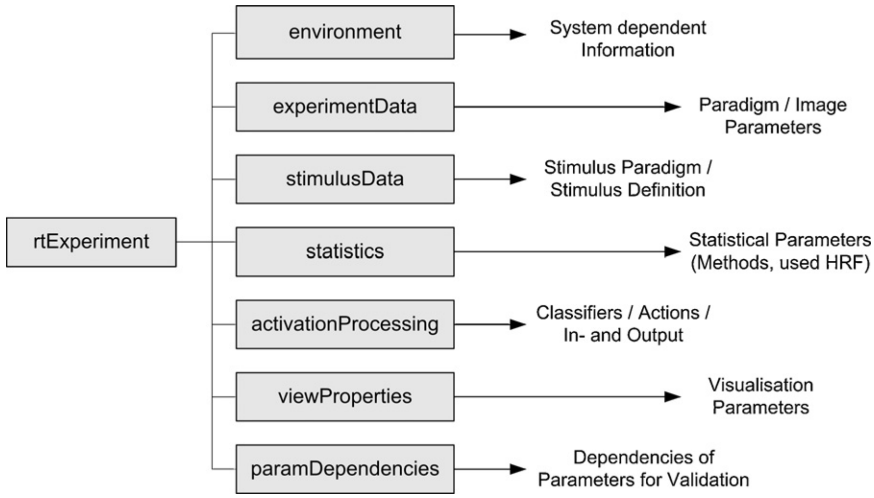

Fig. 3. Structure of the EDL description (version 1.2). Main elements of the central repository and their content as defined in the underlying XML scheme definitions (XSD). The top level elements (colored background) contain additional hierarchically ordered sub-elements. The next level of the EDL code specifies the particular information, e.g. the statistics module defines which statistical method should be used in the described experiment.

In the first line of the EDL code a *correlation analysis* is selected for the real-time statistics procedure. Furthermore, a double-gamma function is used as reference function for the hemodynamic response function (HRF). The reference function is thereby defined by three parameters specifying the overshoot and the undershoot, respectively: the time to peak (tPeak) in s, the full width at half maximum (mWidth) in s, and a scale parameter defining the relation of the peak size of over- and undershoot. To analyze the incoming data for each time step a sliding window with a constant size is selected (see Moench et al., 2007; Hollmann et al., 2007 for a more detailed description). This technique is used if the system has to react to changing brain activation patterns of the subject during the experiment. To preprocess the data a spatial Gauss filter is applied where the size is determined by the code. Changes in the above described parameters can therefore be realized by simply inserting new values via the EDL editor.

The structural correctness can be validated using standard XML libraries. The semantic relations of the given elements have to be checked separately. As the demand of flexibility is better met if the consistency analysis is not hard-coded into the validating application itself, special rules were defined in EDL in the *paramDependencies* section. An example of the relations of single parameters is given in the following code fragment where the dependence between two parameters A and B leads to a change in parameters C and D (up to now for “operation” only simple comparison statements or one of the basic arithmetic operations are allowed):

```
RULE:

IF
  Param-A [operation] Param-B
THEN
  Param-C [operation] Param-D.
```

In this concept all defined rules result in a logical conjunction that is parsed by check routines. One main precondition has to be fulfilled to use the described technique for cross-checks of parameters: All parameters must be uniquely identified. In any XML-based concept this is ensured in the according XSD. More extensive rules can be implemented in a similar manner, but were not needed in the actual version of EDL. Structural validation and the check for semantic relations between the EDL elements are already performed during the editing process in the EDL editor.

The *editor* which is implemented in C++ serves for different tasks. The user can define the experiment parameters conveniently by using just one application for parameters of all connected systems (stimulus presentation, MRI sequence, statistical analysis, and activation processing). In addition, the given data are checked for consistency as well as for potential errors concerning the stimulus design. For example, the application tests whether the stimulus timing is consistent with the timing of events in the statistical analysis. For real-time fMRI experiments the processing speed is important. Therefore the editor allows for simulating the experiment and thereby determines the computational load for all processing steps using simulated data. As another advantageous feature the editor layout is constructed semi-automatically by reading the underlying definition for EDL, the *XSD* file. This ensures that the concept remains flexible if the language definition itself may change in future versions.

2.3. Description of the connected subsystems and information exchange strategies

The concept and the realized solutions for the parameter handling and communication in the subsystems *data acquisition interface*, *stimulus presentation*, *fMRI statistics* and *activation processing* will be outlined subsequently.

2.3.1. Interface to MRI data acquisition

MRI sequence parameters exhibit a complex interdependency from each other partly due to technical restraints (such as between TR and the number of slices) as well as concerning patient safety (e.g. the specific absorption rate to the number of slices, or gradient performance with respect to nerve stimulation). A change of the parameters therefore always requires a check within the context of the underlying measurement sequence. For this purpose the EPI sequence was extended to allow the user to select an EDL file via the vendor’s graphical user interface. This extension is realized within the measurement sequence code, which defines a part of the user interface (the so-called special card) for parameter manipulation in Siemens MR sequences. After selecting an EDL file the sequence reads the MR parameters from the given file and checks them using standard internal solving strategies. If this check fails, because the parameter values are not valid for the given setup the user receives a message and has to change the parameters either via EDL editing or via the direct input at the scanner console. The interface can also be used to write MR parameters into the EDL file. As this interface is specific for each scanner system, it has to be implemented for every measurement facility separately. Currently two versions are implemented for the Siemens scanner software versions VA25A and VB12H.

For our specific vendor platform a development and simulation environment (IDEA) allowed sequence development and test runs offline. For this purpose a special tool of the simulation environment called POET had implemented all checking routines that could therefore be used for our simulation software as well.

2.3.2. Stimulus presentation

The stimulus presentation system was implemented in C++. It uses EDL files as input and presents stimuli according to the defined structure in the EDL section "*stimulusData*". This module manages timing and event logging using trigger signals of the MRI scanner received via parallel port. Subject responses via button devices are transferred via USB ports of the *Stimulus PC* (Fig. 2).

A commercial 3D-engine called "Trinigy" (Trinigy GmbH, Eningen, Germany) (Trinigy, 2006) was used to present virtual 3D environments as well as audio or visual 2D-stimuli. Additionally, an interface to COGENT (COGENT, 2000) was implemented that reads EDL and automatically creates the intended stimulus representation. The correct internal timing is ensured by using output trigger signals from the MRI scanner.

The exact stimulus timing is defined in EDL. Several standard stimulus types such as checkerboards or audio stimuli are already implemented, and can be modified by defining the according parameters in the stimulus section in EDL. More complex user-defined stimuli can be integrated into both systems by using the implemented interfaces of the according applications (the virtual reality environment or the COGENT-based stimulus application).

2.3.3. fMRI statistics

The module to perform the real-time statistical data analysis was implemented in Matlab. It has also a modular structure to ensure high flexibility and to enable a user-friendly extension. The application determines either the BOLD activity in real-time during or after the experiment using an offline analysis. Several standard methods to process fMRI are used, including spatial smoothing, linear detrending, and temporal filters. To extract the brain activation two submodules are available: the first module uses a simple *t*-test, while a second module uses a correlation analysis based on a sliding or growing window mode. The underlying reference vectors can be defined voxel-wise. This enables to examine different brain regions using different hemodynamic response functions accounting for regional differences of the HRF (Aguirre et al., 1998; Handwerker et al., 2004). Together with the *ActivationProcessing* Module (described below) that determines the HRF from an initial measurement, the *Statistics* Module forms the core part of the signal analysis. It parses EDL files to get the essential parameters, such as the paradigm and statistical settings in the EDL section "*statistics*". To disburden the post-experimental SPM-based data analysis EDL furthermore allows to automatically create the SPM *design matrix* (SPM2, 2002).

2.3.4. Activation processing

Similar to the *fMRI statistics* section this component consists of different submodules that can be linked to the measurement process via a module chain. The basic working scheme of such a module is to analyze a given activation state, to classify the particular patterns with respect to some predefined criterion or prior measurements, and to generate an appropriate action. Therefore, the *ActivationProcessing* Module contains several submodules with specialized functions.

(1) The *Activation-Classification-Module* classifies the patterns of activated brain regions. This module works by specifying so-called templates in the first step that depict the activation state of the brain during defined tasks, specified in the according initialization experiments. In the following step, the so-called main experiment, the module correlates these templates with the results of the actual measurements. As a result, the actual pattern may be classified with respect to one of the templates (e.g. "right motor region") as wells as according to changes in extension, amplitude and/or statistical significance with respect to the templates. We used a relevance vector machine for pattern classification (Tipping, 2001).
(2) The *HRF Extraction Module* uses event-related initial experiments to determine the parameters of the subject's individual HRF in different brain regions. Several fitting routines can be used to fit a double- or triple-gamma function to the averaged time courses at different regions of interest. The resulting HRF map may be used as an estimate of the subject- and region-specific HRF for subsequent experiments.
(3) The *Output* and *Receive Module* can be added to the processing chain for the import and export of information into file outputs or into other submodules. These modules are important to allow the communication between different submodules, e.g. between the activation classification and the virtual environment of the *stimulus presentation*. As such, this communication is required to build up a brain computer interface. The modules save or read binary files, which are exactly defined in EDL. The saved information include timestamps (or scan numbers, respectively) of a communication event, and the type and value of a communicated parameter. Thus, all connected systems reading the experiment defining EDL file contain the information which parameters can be found at which time at which location. Output modules may also serve as a device for logging events.

2.4. Performance tests

The performance of the real-time application was tested using simulated fMRI volume data ($128 \times 128 \times 31$ and $64 \times 64 \times 31$). Both datasets contained two artificial activation clusters to verify the computational load for statistics (correlation analysis) and activation classification. Processing speeds were determined with growing window and sliding window designs. In the latter case the classification of the actually derived activation state was included. Mean processing times were derived averaging the computational load for 500 volumes respectively at an Intel Pentium IV, 3.0 GHz, and 2 GB RAM. Since the overall performance also depends on the data acquisition and image reconstruction performance of the MR system the performance parameters must be determined for each scanner site and measurement sequence separately.

2.5. Usability tests

To test the usability of the approach five neuroscientists were asked to create a simple fMRI experiment using a checkerboard stimulus. At a scale of 1–10 (no knowledge in fMRI to perfect knowledge) three participants described their fMRI skills with 9, one with 8, and one with 6. All participants were familiar with SPM2 (SPM2, 2002) and Presentation 10.0 (Presentation, 2007).

The task was to build up an fMRI experiment for the Siemens 3 T (VA25A) scanner with given MRI measurement parameters and a jittered visual stimulus design (8 Hz flickering checkerboard) of overall 20 blocks (10–13 images per block). The stimulus design was either implemented with the *EDL editor* or with *Presentation*, respectively. The second part of the test consisted of creating the

SPM design matrix. The last step was to prepare the measurement parameters at the MRI scanner user interface. For the simulation of the user interaction with the scanner user interface (Siemens Syngo) the POET tool was used, which is part of the Siemens IDEA sequence development environment.

These three steps were conducted under the control of the investigator. For each step the exact handling time was logged, and potential design errors were analyzed retrospectively. After an introduction to the *EDL editor* and user guidance how an EDL file is loaded into the custom modified EPI-sequence (*data acquisition interface*) the described step-by-step procedure was applied. Afterwards the participants rated the usability of the EDL approach by answering questionnaires under three aspects: (1) efficiency; (2) comprehensibility/user friendliness; (3) error-proneness. The rating scale was 1 (low) to 10 (high).

2.6. BCI experiments

The complete system was evaluated by implementing a BCI as a proof-of-principle experiment. Two subjects (one male, one female, aged 29 and 27, both right-handed) were examined after written consent according to the local ethics committee.

On both MRI sites the imaging protocol consisted of gradient echo (GE) EPI sequences for BOLD imaging (3 T: repetition time (TR) 2 s, time to echo (TE) 30 ms, flip angle 90°, 31 slices, axial slice orientation, matrix 64×64, spatial resolution 3.4 mm × 3.4 mm × 4.5 mm; 7 T: TR 2 s, TE 21 ms, flip angle = 90°, 16 slices, 64×64 matrix, spatial resolution 3.4 mm × 3.4 mm × 6 mm). The lower number of slices at 7 T was due to restrictions concerning the specific absorption rate (SAR) using the vendor EPI sequence.

The experiment consisted of two parts. In the first part functional areas were localized using a motor paradigm (finger tapping left and right hand) as well as a mental calculation paradigm. Therefore, three templates for the according different brain activation patterns were determined. This initial measurement consisted of a block of 42 images for each condition: 3 times [5 scans active–9 scans baseline]. After the initialization experiment, the templates for each condition were determined semi-automatically and stored as reference templates. In the next step, the main experiment, these reference templates were used to classify the actual brain activation achieved with a sliding window of a constant size of 12 images: [2 scans ignore–3 scans active–7 scans baseline]. The commands “start” and “stop” were presented with the described stimulus application via scanner-compatible headphones.

During the main session the volunteers had to navigate through a virtual reality 3D maze by using different brain activation patterns. By default, the player moved along the aisles of the maze and stopped at the crossings. There the subsequent motion was decided according to the brain activation classification result: movement of the left hand led to a 90° turn to the left; movement of the right hand led to a 90° turn to the right; mental calculation resulted in moving forwards.

For validation, the volunteers indicated their intended decision by pressing buttons after the active period of each run. These intended decisions and the classification results were logged on the *Stimulus PC* for a subsequent analysis of the classification quality. The volunteers passed several “training sessions” to optimize their performance during the experiments.

3. Results

3.1. Performance tests

Table 1 depicts the mean processing times for correlation analysis with and without classification at the simulated datasets and shows that the computational load for a volume of $128 \times 128 \times 31$ elements can reach the critical limit of 2 s repetition time. Considering these results the prior simulation of the experiment is a fundamental step for the planning of an rtfMRI experiment. In the most demanding test runs the memory (RAM) load was about 300 MB. This was mainly caused by voxel-wise reference vectors and holding intermediate results for the correlation analysis.

Table 1
Processing times (s) for a single volume using sliding and growing window correlation analysis as function of volume dimensions

	$64 \times 64 \times 31$ voxel	$128 \times 128 \times 31$ voxel
Growing window	0.63	1.78
Sliding window (no classification)	0.51	1.52
Sliding window (including classification)	0.72	2.13

3.2. Usability tests

For the user evaluation of the whole system three benchmarks were used: (1) time required to create an exemplary experiment; (2) number of errors in stimulus design, design matrix, or MRI parameters; (3) subjective judgment of the participants.

Table 2 depicts the mean time needed for processing the given task, split in the three consecutive steps: Stimulus Creation, Creation of Design Matrix, and Preparing MRI Parameters.

As shown, the EDL approach substantially saved time during the preparation of an fMRI experiment. An important part that contributed to this result was the required programming effort using *Presentation* for stimulus presentation which is much more complex than the stimulus design definition in the EDL editor. Furthermore, the creation of the SPM design matrix is fully automated in the EDL approach, while it requires careful effort in SPM. However, due to the high expertise of the test persons only one error in stimulus design was found at the *Presentation* code of one participant, while no errors were made using the EDL approach.

As already stated, the interface to the MRI scanner is an important step to improving usability, because it allows an automated import and export of measurement parameters at the scanner. This benefit was confirmed by the volunteers.

The averaged ‘marks’ given by the participants were: efficiency: 8 (8 for Presentation/SPM approach), comprehensibility/user friendliness: 6.4 (4 for Presentation/SPM approach), error-proneness: 8.2 (3.5 for Presentation/SPM approach). One major problem described by the volunteers was a somewhat lacking intuitive user interaction of the EDL editor. This will be taken into account in future versions of our framework.

Table 2
Mean processing times (in min) for the single workflow steps in the experiment design process comparing the standard approach (using *Presentation* and *SPM2*) and the EDL-concept

	Mean time (S.D.)	
	Presentation/SPM2	EDL approach
Stimulus Creation	22.3 (14.1)	16.4 (3.2)
Creation of Design Matrix	2.2 (0.5)	Automated
Preparing MRI Parameters	3.4 (0.7)	1.2 (0.1)
Sum	27.9	18.1

The creation of design matrices using EDL was automated and required no further user interaction.

3.3. BCI experiments

As one of the advantages of the system the real-time data analysis immediately allowed to detect movement artifacts during the experiments. These were typically depicted as false positive activations localized on tissue borders. They were minimized by giving the subject the respective instructions during the measurement. In the main experiments a high number of correct classifications were reached for the different conditions and for both scanner systems. In a total of 80 single runs per volunteer 97% (left tapping), 99% (right tapping), and 91% (mental calculation) were correctly identified. The classification was very reliable on both scanners (mean BOLD signal was 3.3% at 3 T and 6.8% at 7 T). The high classification quality enabled the subjects to move easily through the virtual maze.

4. Discussion

The developed concept of a centralized parameter management was successfully implemented and tested in rtfMRI experiments. Using high field MRI for a brain computer interface the subjects were able to navigate through a virtual reality maze by evoking several activation patterns. This task required a prior determination of activated areas, which was realized in initial experimental runs. The results were stored as templates, and subsequently used in the main experiment for classification.

In this main experiment, the statistical analysis used a different setting (sliding window instead of growing window, different stimuli, and different block length). However, the user only had to select a particular EDL file of the main experiment which by itself pointed to the EDL files of the prior experiments and contained all information for the execution of the main experiment.

Although in principle the described concept could also be implemented in different software realizations, the development of an XML-derived language allowed to benefit from several advantages, among them the use of sophisticated libraries to check the syntax and to correct the parameter definitions in EDL which are based on the language definition (XSD).

During the planning process and the realization of the measurements the concept of storing the parameters in EDL files proved to be advantageous in additional aspects:

1. The setting of the experimental parameters using the editor was user-friendly and reliable due to the ongoing consistency checks while editing the EDL definition.
2. The description of the MR parameters in EDL, and their transfer via the interface to the Siemens 3 T and 7 T scanners was helpful to increase the usability. This interface shortens the processing time and helps avoiding input errors.
3. The option to simulate the experiment after having created all required EDL files proved to be an important help to develop real-time experiments and to test the experimental workflow prior to the experiment with respect to a coherent parameter setting.
4. By implementing interfaces to other applications EDL helped to ease the data analysis process. As one example the custom-made interface to SPM may be quoted. The automated generation of the SPM design matrix based on the EDL information shortened the post-processing time for functional MRI data with SPM substantially.
5. Another important advantage is the standardized connection of different subsystems of the measurement process. The well-defined communication processes that included which components receive or send which information, in which format, or at which time enabled the realization of dynamic paradigms. By incorporating EDL a software application is enabled to check automatically whether all subsystems are able to meet the demands for the defined communication within their given parameter sets. This is enabled by using automatically generated simulation setups.

Although various software solutions for real-time applications offer a part of functionalities the complete set of functionalities of our framework is not met until now. Popular and state-of-the-art solutions for rtfMRI are e.g. the commercial application Turbo Brain Voyager (Goebel, 2002–2006) or the free tool FIRE (Gembris et al., 2000; Posse et al., 2001, 2003). However, these applications are specialized rtfMRI approaches and do not incorporate other measurement systems in a standardized manner.

A step towards generalization was offered by Weiskopf et al. (2004b). In this approach a connection between the software system Turbo Brain Voyager and the BCI of the working group was established. Nevertheless, this connection is done via software code and is restricted to the commercial Turbo Brain Voyager software, and as such not usable for the configuration of other rtfMRI systems, stimulus presentation, or MRI scanner-system, respectively.

Other software systems support users in generating the paradigm setting and stimulus presentation (see Stahl, 2006 for a review). Two examples for these well-established tools are DirectRT (DirectRT, 2004) or E-Prime (E-Prime, 2002). Again, these subsystems create a well-formed setup inside the stimulus presentation component of an experiment, but a consistent overall parameter setup is not assured, and has to be realized by the user.

A few solutions for fMRI/rtfMRI exist trying to overcome these problems by including the complex task of a unified experimental control (Voyvodic, 1999; Smyser et al., 2001).

Voyvodic combined stimulus presentation, rtfMRI statistics, and recording of physiological parameters in a unique system by using a central communication scheme to ensure the correct timing of the components. A main feature of this approach is the use of central parallel processes for controlling the connected systems (stimulus presentation and statistics) running on a real-time processor. The paradigm for the statistical analysis is directly created by the stimulus presentation. Thereby the correct timing of all components is ensured and there is no need for cross-checks of stimulus presentation and statistical parameters. The latter point may be limiting the flexibility in statistical analysis. Although the timing of the workflow of the subsystems is ensured, no central parameter description enabling pre-evaluation and simulation of experiments is used, and an expansion by other systems (e.g. different statistics) is not part of the concept.

Smyser presented an approach for rtfMRI that incorporated time-dependent parameters (physiological data, stimulation trigger, subject responses, etc.) and linear regression methods. Here, the timing signals of the mentioned subsystems were recorded including the corresponding time-stamps. The main intention was to present a convenient way for time-coherent measurements. For that purpose every log signal of the subsystems was stored in CVIO files (cardiovascular I/O). These files are used for the system's output but not for the input of several subsystems, e.g. the stimulus presentation signals were recorded, but the definition of the design was not realized in a revisable way.

In both approaches sophisticated solutions for a fast communication and/or recording of time-dependent parameters were developed, but the unification of the systems is not accomplished by creating a conjoint parameter space but only by ensuring exact timing (which is of course essential for real-time fMRI). However, those methods do not separate the parameter representation and the systems implementation, thus the experiment parameter space cannot be easily expanded, e.g. by the MRI measurement parameters, which are implicated in no existing approach except for the presented one.

EDL is a realization of a more abstract concept where the description of all parameters is represented in any of the discussed subsystems in an encapsulated way. Thus EDL opens the possibility to hold all parameters and possibly to translate them into the internal representation of any open source system by implementing interfaces (as realized with COGENT, 2000 and SPM).

This ensures that for example the automated generation of measurement protocols as well as the automated volunteer management (such as developed by Yule and Cooper, 2003) are easily to add to the whole experiment process. The next version of EDL will include subject-specific data to cope with that task.

Other systems such as the commercial software Eloquence/IFIS (Eloquence, 2006) offer a complete solution for fMRI including Brain Voyager and Turbo Brain Voyager for fMRI/rtfMRI statistics, a software suite for stimulus design, and several hardware components. As both are commercial applications the source code is not available and no bidirectional interfaces for further activation analysis and stimulus-adaption are described, which limits the extensibility and the usage for dynamic experiments. For example it is not possible to automatically create an SPM design matrix or to adapt MRI measurement parameters for experiments implemented in Eloquence/IFIS.

Despite the advantages of our concept some parts still have to be improved. The presented subsystems for real-time statistics and activation analysis (including classification) may be extended to include a general linear model (GLM) as already used in real-time applications (Bagarinao et al., 2003; Nakai et al., 2006). This might render the statistical analysis more stable and powerful. Furthermore the EDL editor should exhibit a more intuitive user interaction, e.g. “drag and drop” functionality as suggested by the results of the usability study.

Additionally, our system does not yet include recording and handling of physiological parameters as this was not yet crucial for our experiments.

Other critical details concerning the description language may also be of importance:

1. The whole approach is very flexible concerning the expansion, but new applications and thus parameters require the definition of the new elements in the language itself (XSD) as well as the implementation of the mutual parameter dependencies in EDL.
2. The power of the approach is limited to a description that can be expressed in EDL. If this is not the case the according parameter description fact will not be part of the evaluation routines and thereby the generalization fails. Therefore the development of EDL is not trivial and will need ongoing effort in the future.
3. The EDL interfaces to the local hardware (Siemens 3 T and 7 T MRI scanners) are essential parts of the overall concept especially concerning the Siemens scanner software environment for checking the MRI parameters in an EDL file. For other vendor-specific environments users may have to implement own interfaces for other local hardware, where the functionality will depend on these vendor-specific functions.

The description and control of experiments with EDL is of course not restricted to fMRI measurements. The same approach may for example be used for other neuroimaging experiments such as Electroencephalography (EEG), Magnetoencephalography (MEG), functional Near Infrared Imaging (fNIR), or any modality where different systems build up a complex measurement environment. However, connecting other modalities, stimulus presentation programs or statistical applications requires then a programming of the according interfaces to EDL.

An even more advanced application may be realized as EDL can be adapted to standardize the communication between all kinds of systems that are involved in the measurement process. Consequently this approach could be expanded to the standardized communication between several scanning sites, such as connecting the 3 T and 7 T scanners as it was already implemented and validated in a proof-of-principle experiment.

The basic definition of EDL and the software components of the framework will be available upon request as soon as the final modifications concerning user friendliness are accomplished.

5. Summary

A new concept of a consistent parameter description and parameter control was successfully implemented and tested in real-time high field fMRI experiments. An XML-based description language was developed that uniquely represented all parameters of the systems involved in an fMRI measurement. By using a centrally stored EDL parameter description file in combination with flexible software applications a consistent experimental setup and well-defined communication streams were realized. The inherent interdependency check with respect to semantic and structural consistency facilitated the planning and realization of the real-time experiments as well as the data evaluation. The approach is therefore another important step towards the concept of *intelligent imaging*, i.e., incorporating as much information of the data acquisition and analysis process into meta information that can be processed by the systems themselves with as few user interaction as possible. EDL could support a standardized approach to unify several software systems in rtfMRI and fMRI.

References

Aguirre GK, Zarahn E, D'Esposito M. The variability of human, BOLD hemodynamic responses. NeuroImage 1998;8:360–9.
Bagarinao E, Matsuo K, Nakai T, Sato S. Estimation of general linear model coefficients for real-time application. NeuroImage 2003;19:422–9.
COGENT, 2000. Laboratory of neurobiology. London: University College. http://www.vislab.ucl.ac.uk/Cogent2000/index.html.
DeCharms RC, Christoff K, Glover GH, Pauly JM, Whitfield S, Gabrieli JD. Learned regulation of spatially localized brain activation using real-time fMRI. NeuroImage 2004;21:436–43.
DeCharms RC, Maeda F, Glover GH, Ludlov D, Pauly J, Soneji D, et al. Control over brain activation and pain learned by using real-time functional MRI. PNAS 2005;102(51):18626–31.
Direct RT. Empirisoft, New York; 2004. http://www.empirisoft.com.
Eloquence. Intermagnetics, Orlando; 2006. http://www.igc.com.
E-Prime Psychology Software Tools. Pittsburgh, PA; 2002. http://www.pstnet.com.
Gembris D, Taylor JG, Schor S, FringsW, Suter D, Posse S. Functional magnetic resonance imaging in real time (FIRE): sliding-window correlation analysis and reference-vector optimization. Magn Reson Med 2000;43:259–68.
Goebel R. Turbo Brain Voyager; 2002–2006. http://www.brainvoyager.com/TurboBrainVoyager.html.
Handwerker DA, Ollinger JM, D'Espposito M. Variation of BOLD hemodynamic responses across subjects and brain regions and their effects on statistical analyses. NeuroImage 2004;21:1639–51.
Hollmann M, Moench T, Tempelmann C, Bernarding J. A unified approach for fMRI-measurements used by a new real-time fMRI analysis system. Proc Bildverarbeitung Med 2007:56–60.
Matlab—The Language of Technical Computing R2006b. The Mathworks Inc.; 1984–2006. http://www.mathworks.com/products/matlab/.
Moench T, Hollmann M, Bernarding J. Real-time fMRI based activation analysis and stimulus control. Proc SPIE Med Imag 2007;6511:651114.
Nakai T, Bagarinao E, Matsuo K, Ohgamic Y, Kato C. Dynamic monitoring of brain activation under visual stimulation using fMRI—the advantage of real-time fMRI with sliding window GLM analysis. J Neurosci Methods 2006;157:158–67.
Posse S, Binkofski F, Schneider F, Gembris D, Frings W, Habel U. A new approach to measure single-event related brain activity using real-time fMRI: feasibility of sensory, motor, and higher cognitive tasks. Hum Brain Mapp 2001;12:25–41.
Posse S, Binkofski F, Gao K, Habel U, Rosenberg D, Moore G, et al. Real-time fMRI of temporolimbic regions detects amygdala activation during single-trial self-induced sadness. NeuroImage 2003;18:760–8.

Posse S, May G, Villarreal G. Online modeling of the temporal dynamics of the anxiety response in post-traumatic stress disorder (PTSD) using real-time functional mri. In: Proceedings of the 13th Human Brain Mapping; 2007.

Presentation—Version 10.0. Neurobehavioral Systems; 2007. http://www.neurobs.com/presentation/.

Smyser C, Grabowski TJ, Frank RJ, Haller JW, Bolinger L. Real-time multiple linear regression for fMRI supported by time-aware acquisition and processing. Magn Reson Med 2001;45:289–98.

SPM2—Statistical Parametric Mapping Version 2002. UCL; 2002. http://www.fil.ion.ucl.ac.uk/spm/.

Stahl C. Software for generating psychological experiments. Exp Psychol 2006;53(3):218–32.

Tipping ME. Sparse Bayesian learning and the relevance vector machine. J Mach Learn Res 2001;1(3):211–44.

Trinigy 3D Engine 6.01. Trinigy GmbH; 2006. http://www.trinigy.net/.

Voyvodic JT. Real-time fMRI paradigm control, physiology, and behavior combined with near real-time statistical analysis. NeuroImage 1999;10:91–106.

Weiskopf N, Scharnowski F, Veit R, Mathiak K, Goebel R, Birbaumer N. Physiological self-regulation of regional brain activity using real-time functional magnetic resonance imaging (fMRI). J Physiol (Paris) 2004a;98:357–73.

Weiskopf N, Mathiak K, Bock SW, Scharnowski F, Veit R, Grodd W, et al. Principles of a brain–computer interface (BCI) based on real-time functional magnetic resonance imaging (fMRI). IEEE TBE 2004b;51(6):966–70.

Xerces-C++ 2.70. XML-Library; 2001–2005. http://xml.apache.org/xerces-c/.

XML—Extensible Markup Language Version 1.0. W3C; 1996–2007. http://www.w3.org/XML/.

Yoo SS, Fairneny T, Chen NK, Choo SE, Panych LP, Park H, et al. Brain–computer interface using fMRI: spatial navigation by thoughts. Neuroreport 2004;15:1591–5.

Yule P, Cooper R. Express: a web-based technology to support human and computational experimentation. Behav Res Methods Instrum Comput 2003;35(4):605–13.