

# Ontology-Driven Toolset for Audio-Visual Stimuli Representation in EEG-Based BCI Research

Konstantin Ryabinin<sup>1</sup>, Svetlana Chuprina<sup>1</sup> and Ivan Labutin<sup>1</sup>

<sup>1</sup>Perm State University, Bukireva Str., 15, Perm, 614068, Russia

## Abstract

In the last decade, the recent advances in software and hardware facilitate the increase of interest in conducting experiments in the field of neurosciences, especially related to human-machine interaction. There are many mature and popular platforms leveraging experiments in this area including systems for representing the stimuli. However, these solutions often lack high-level adaptability to specific conditions, specific experiment setups, and third-party software and hardware, which may be involved in the experimental pipelines. This paper presents an adaptable solution based on ontology engineering that allows creating and tuning the EEG-based brain-computer interfaces. This solution relies on the ontology-driven SciVi visual analytics platform developed earlier. In the present work, we introduce new capabilities of SciVi, which enable organizing the pipeline for neuroscience-related experiments, including the representation of audio-visual stimuli, as well as retrieving, processing, and analyzing the EEG data. The distinctive feature of our approach is utilizing the ontological description of both the neural interface and processing tools used. This increases the semantic power of experiments, simplifies the reuse of pipeline parts between different experiments, and allows automatic distribution of data acquisition, storage, processing, and visualization on different computing nodes in the network to balance the computation load and to allow utilizing various hardware platforms, EEG devices, and stimuli controllers.

## Keywords

Ontology Engineering, Stimulus Representation, Brain-Computer Interface, EEG, Visual Analytics

## 1. Introduction

Electroencephalography (EEG) is a non-invasive method to study brain activity measuring the electrical potentials on the scalp skin. Traditionally, EEG relates to healthcare leveraging brain injury diagnostics, as well as to neurophysiological research that targets the discovery of the physiology behind human cognition. Currently, with the evolution of digital signal processing, EEG techniques enter the area of human-computer interaction, introducing the new interaction paradigm called brain-computer interfaces (BCIs). As a result, in the last decade, several hardware and software platforms have been developed to conduct EEG-based research and analyze its results. However, the common problem of the most popular solutions is low-level adaptability. This means one should manually reprogram individual modules of existing

---

*GraphiCon 2021: 31st International Conference on Computer Graphics and Vision, September 27–30, 2021, Nizhny Novgorod, Russia*

✉ kostya.ryabinin@gmail.com (K. Ryabinin); chuprinass@inbox.ru (S. Chuprina); i.a.labutin@yandex.ru (I. Labutin)

🆔 0000-0002-8353-7641 (K. Ryabinin); 0000-0002-2103-3771 (S. Chuprina); 0000-0001-6858-1479 (I. Labutin)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

platforms, or create ad-hoc software (and sometimes even hardware) adapters to build an EEG data acquisition pipeline for a particular experiment.

In the present work, we tackle this problem by proposing a unified ontology-driven hardware and software system that enables the representation of auditory and visual stimuli along with recording, processing, and visual analysis of EEG data. This system relies on the SciVi client-server visual analytics platform (<https://scivi.tools>) developed in our previous work [1]. This platform consists of an extensible set of plugins managed by means of a SciVi knowledge-driven mechanism. In the knowledge base, ontological descriptions of the plugins are stored. The built-in ontology reasoner traverses the SciVi ontologies to automatically generate a graphical user interface and interoperation interface for each plugin, as well as manages the invocation of the plugin's executable modules and distributes their working process across the available computing resources in the network.

SciVi platform provides two levels of adaptation to the particular data mining tasks. The first adaptation level is available for the platform administrator, who extends the SciVi ontologies and thereby adds new plugins implementing new data analytics methods. This is the way SciVi is adapted to solving tasks in particular application domains. The second adaptation level is available for the end users, who declare a particular data mining pipeline using visual programming language based on data flow diagrams (DFDs). Composing a flow chart of available operators (each one corresponding to the particular SciVi plugin) and data links, end users can implement particular data extraction, transformation, and load (ETL) algorithms, as well as related data filtering and visualization steps. In this way, SciVi is adapted to solving concrete data analysis problems in a specific application domain.

The present work is devoted to new capabilities of the SciVi platform ensuring audio-visual stimuli representation in EEG-based BCI experiments and visual analytics of their results.

The key contributions of our paper are the following:

1. Introducing a new method for integrating stimuli presentation platforms with experimental environments in a unified way by the means of ontology engineering.
2. Presenting a new way to compose and control a BCI experiments pipeline with the help of an ontological description of the neural interface used.

## 2. Related Work

Thanks to the comprehensive review of current status, challenges, and possible solutions of EEG-based brain-computer interface provided by M. Rashid et al. [2], here we present only a short review of the most widely acknowledged and popular free/open-source and commercial EEG-based BCI systems.

BioSig [3] (<http://biosig.sourceforge.net/>) is one of the oldest MATLAB-based tools for building BCI-enabled applications and conducting neuroscientific studies. It has a very wide set of available data processing algorithms, but it only provides very basic visualization methods via its SigViewer subproject (<https://github.com/cbrnr/sigviewer>) and it is also designed for offline data analysis.

Another very distinguished platform for EEG analysis is BCI2000 [4] (<https://www.bci2000.org/>). It is a highly modular and robust cross-platform solution for data collection, signal

acquisition, and stimuli presentation in real-time. It supports a wide range of EEG devices, has very good documentation, and a welcoming community. However, it lacks modern signal processing and machine learning algorithms and is designed more towards providing an “out-of-the-box” experience for typical scenarios.

OpenVibe [5] (<http://openvibe.inria.fr/>) is a relatively new tool for neuroscience experiments and is geared towards non-programmers. It has a modular architecture and provides the user with DFD-like diagrams to build the experiments pipeline. OpenVibe is very user-friendly both in terms of user interface and documentation and supports a wide range of hardware. But it also lacks sophisticated adaptive signal processing and machine learning methods, and it is very hard to extend it due to its complex architecture. It is written in C++; the two main platforms it targets are Windows and GNU/Linux.

g.BSanalyze (<https://www.gtec.at/product/gbsanalyze/>), just as BioSig, is also a MATLAB plugin that allows the user to analyze recorded biosignal in a highly customizable and flexible interface, featuring many advanced algorithms. It is a commercial solution though and is a part of a complete “turnkey” solution for deploying a neuroscientific laboratory. It is also an offline solution.

BCILAB [6] (<https://github.com/sccn/BCILAB>) is also an Octave/MATLAB toolbox for conducting neuroscientific studies. It has one of the largest collections of signal analyzing and processing methods available, supports both online and offline modes, and can easily be extended with plugins. But it has not been actively developed since 2017, and it has a very complex internal architecture, making it quite tough to maintain on your own.

FieldTrip [7] (<https://www.fieldtriptoolbox.org/>) is yet another MATLAB plugin aimed at MEG and EEG analysis. It is very young and is being actively developed, has basic processing modules for signal processing and visualization, but is still under heavy construction.

xBCI [8] (<http://xbci.sourceforge.net/>) is also a tool for building BCI and conducting online neuroexperiments. Like OpenVibe, it features a DFD-like GUI pipeline editor, making it suitable for use by non-programmers, and employs an extensible plugin-based architecture. Unfortunately, it has not been updated since 2008.

PyFF [9] (<http://bbci.de/pyff/>) is a Feedback Framework written in Python. Its main purpose is to simplify creating neurofeedback applications by utilizing a relatively simple but still general-purpose programming language Python as its core scripting engine. PyFF is very convenient and easy to use for IT specialists, but the project was not updated for 6 years.

SNAP (<https://github.com/sccn/SNAP>) is the Simulation and Neuroscience Application Platform based on the Panda3d computer game engine and aims to bring complex human-computer interaction into the field of neuroscience. It uses Python as a scripting language and is easily extensible with custom plugins. Unfortunately, as of 2021, the platform has not been updated in 8 years and falls behind significantly in terms of modern infrastructure.

E-Prime is a software platform “designed to facilitate the conception of any experiment that uses a computer as an interface between the subject and the experimenter” [10]. It follows the paradigm of an integrated research environment, supporting the study from its idea through design and conduct to the results processing steps, and features many high-level tools, including a toolset for stimuli presentation. But its course towards an “all-in-one” solution, proprietary nature, and the lack of modularity poses a problem in case of the need for integration.

DMDX [11] is a tool for stimulus presentation in linguistic experiments with the stress on very

precise and accurate timing of stimuli. It is very stable and mature and is widely popular, but it is proprietary, closed-source, and lacks support for any platforms besides desktop Windows. It also was not updated in recent years.

PsychoPy [12] is a free, open-source, and cross-platform toolbox for conducting experiments in behavioral sciences. It is very flexible, but nonetheless, the only extension method it supports is Python scripting.

BOLDSync [13] is a stimulus presentation framework designed specifically for neuroscience studies. It employs client-server architecture and uses a VLC media player for stimulus presentation. It is open-source, but it is based on MATLAB and not really designed to be extensible, and primarily targets functional magnetic resonance imaging studies.

ViSaGe (<https://www.crsiitd.com/tools-for-vision-science/visual-stimulation/visage/>) is a stimulus presentation solution that is quite unique from the other tools in our list. It is an integrated hardware and software system for stimuli presentation that allows precise control over the timing, color, and luminance of the visual stimuli presentation. On the other hand, ViSaGe relies on MATLAB to enable integration with third-party systems, quite pricey and by design requires external hardware.

Psychtoolbox [14] is yet another specimen from the family of MATLAB plugins. Its goal is to provide researchers with a set of utility functions and tools to be used during stimulus presentation. It is very mature and still ongoing, but its aim to be a MATLAB plugin drastically affects its integrability.

In summary, there are a lot of popular, mature, and robust tools for conducting neuroexperiments in general and presenting various types of stimuli, but to the best of our knowledge, there are none that can provide seamless and unified integration with other software and hardware systems.

The authors of [2] mention that “a general BCI standard is currently the main issue. Most of the studies on BCI have used different evaluation metrics on their own as per their convenience without any uniformity, which makes it difficult to choose the most efficient method, especially for new researchers in this field.”

Our paper presents an original ontology-driven solution to tackle this problem.

### **3. Stimuli Representation Tools**

#### **3.1. Stimuli Representation Pipeline**

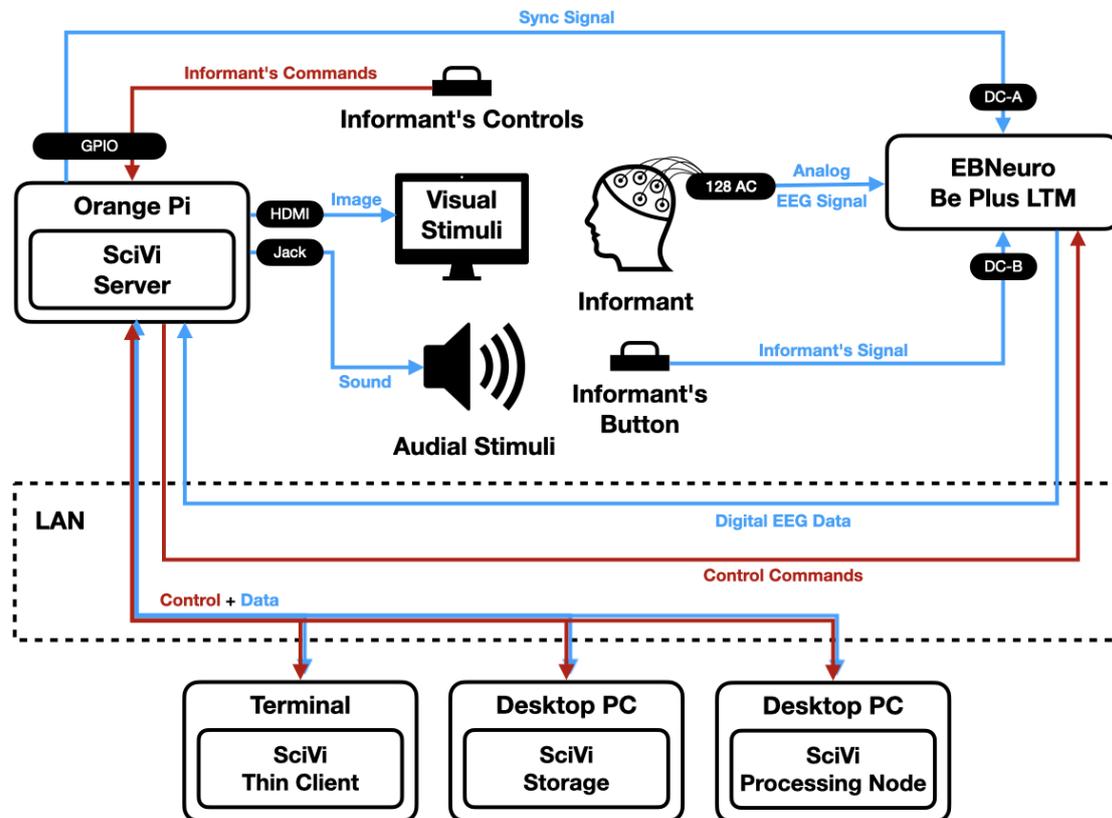
To get the meaningful data for further analysis (involving both machine learning algorithms and visual analytics performed by experts), stimuli representation should be accurately synchronized with the EEG signal. Often, the presence/absence of a particular stimulus is encoded as a high/low signal level in the special channel of EEG recording along with the other channels, which represent brain activity. The rising edge of the signal in this special channel should perfectly match the time the stimulus appears. The falling edge should correspondingly match the time the stimulus disappears.

To achieve the needed time-scale accuracy, special hardware solutions are involved, for example, photo-sensors mounted on the monitor that shows the visual stimuli. These sensors emit electrical impulses whenever a new stimulus is shown on the monitor, and the impulses

are recorded by an EEG device along with the signal from the headset. This kind of registration system is however limited to unimodal stimuli (visual only). A more flexible way is to emit electrical impulses directly by the computer that presents stimuli.

To minimize the output lag of the impulses emitted we propose using a single-board microcomputer with general-purpose input-output (GPIO) pins available. The most popular microcomputers are Raspberry Pi and Orange Pi. In the present work, we adopted Orange Pi PC Plus that has 28 different GPIO pins along with the power lines of 3.3 V and 5 V, as well as built-in Ethernet and WiFi adapters. It is based on the H3 Quad-Core Cortex-A7 CPU, has 1 GB DDR3 RAM and Mali400MP2 GPU supporting OpenGL ES 2.0. Although the overall performance of this computer is fairly low (compared to desktop computers), it is enough to present different kinds of stimuli, including text, images, animations, videos and sound. Moreover, Orange Pi can be transparently integrated with tangible user interfaces [15], allowing to involve haptics modality.

The schema of the stimuli representation pipeline proposed is shown in Fig. 1.



**Figure 1:** Stimuli representation pipeline based on the SciVi platform.

As shown in this figure, SciVi Server hosted on the Orange Pi PC Plus single-board microcomputer represents visual (by the monitor connected via HDMI port) and auditory (by the speaker or headphones connected via a mini-jack port) stimuli to the Informant.

The electrical signal from the Informant’s headset transmitted over up to 128 analog channels (128 AC) is being registered by an EBNeuro Be Plus LTM device that contains an analog-to-digital converter, an amplifier, and a communication module to stream the digitized signal over the local area network (LAN). This signal is being received by the SciVi Server that incorporates software logic of parsing EBNeuro network packages and controlling the EBNeuro device state (for this, a self-written driver is used). The SciVi Server acts as a proxy for the SciVi Storage, where the EEG data are saved, and for the SciVi Processing Node, where the data are filtered, clusterized and classified. The experiment director accesses the SciVi Server via a Terminal (arbitrary desktop computer, laptop, or mobile device) running the SciVi Thin Client.

Whenever the stimulus appears, SciVi Server uses Orange Pi GPIO pins to send a synchronization signal that is received via a special DC-A input of the EBNeuro device and recorded along with EEG data.

If the experiment requires actions from the informant, two control circuits are provided: Informant’s Button (the button that emits square-shaped signal recorded by EBNeuro device through special DC-B input) and Informant’s Controls (arbitrary Joystick-like controller connected to the Orange Pi via GPIO pins). Informant’s Button may be used to store the feedback from the informant. For example, an informant can indicate whether he/she has imagined some situation that is studied in the particular experiment. Informant’s Controls are used to give the informant potential control over the stimuli presented, for example, to navigate between them. Currently, pushbuttons are used, but in the future more specific hardware interfaces can be adopted, including tangible ones [15].

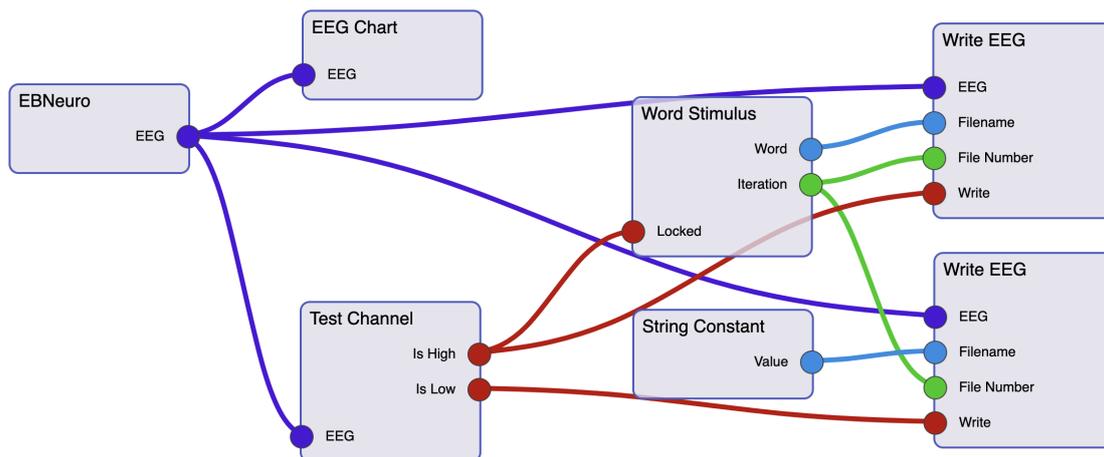
SciVi Server, SciVi Storage, and SciVi Processing Node are implemented in Python, having a lot of common code related to network discovery and communication (that is mainly based on the WebSocket protocol). SciVi Server relies on the Flask framework (<https://flask.palletsprojects.com/>) to allow HTTP-based communication with Web clients. Server-side plugins are mainly implemented in Python too, but some of them use native libraries written in C++. SciVi Processing Node relies on SciPy (<https://www.scipy.org/>), MNE (<https://mne.tools/>), and scikit-learn (<https://scikit-learn.org/>) libraries to perform machine-learning-based processing and analysis of EEG data. SciVi Thin Client is written in JavaScript relying on HTML5 and CSS3.

### 3.2. Data Flow Declaration

As mentioned above, ETL-, filtering-, and visualization pipelines in SciVi are declared with the help of DFDs composed of high-level operators. Each operator has its ontological description [1]. An example of the DFD declaring a presentation of words as a specific type of visual stimuli is shown in Fig. 2.

In this DFD, the “EBNeuro” operator is responsible for communicating with the EBNeuro Be Plus LTM EEG device. Implementation of this operator uses a self-written device driver running on the server-side. It sends control commands to the EBNeuro device and receives the EEG data stream.

“EEG Chart” is a client-side visualization tool that draws received EEG data stream as a line chart (see Section 4 for details). It is worth noting, that the data link from the “EBNeuro” operator to the “EEG Chart” operator incorporates automatic data serialization and marshalling on the SciVi Server, as well as data receiving and deserialization on the SciVi Thin Client. The



**Figure 2:** SciVi DFD declaring a presentation of word stimuli.

mechanisms of automatic marshalling based on DFD make data transfer inside SciVi transparent for the user: the user just declares the sequence of operators, which should be applied to the data, and does not worry about where a particular operator is running and how exactly it receives/transmits the data.

The “Word Stimulus” operator is responsible for presenting the sequence of words on the screen. It has corresponding settings, which allow to set up the list of words, the duration of showing individual words, and the number of times (iterations) the list of words is presented. Each word appears on screen, stays for a given time, then disappears, and the screen stays black for a while; after that, the next word appears. This process is repeated  $n \cdot m$  times where  $n$  is a number of words and  $m$  is a number of iterations requested. Simultaneous with the showing of each individual word, the synchronization signal is generated using the Orange Pi GPIO pin, which is connected to the DC-A input of the EBNeuro device. When the word is shown, the level of the synchronization signal is set to high, otherwise, it is set to low.

“Test Channel” operator reports whenever the voltage level in the given channel (DC-A in our case) is high (above a given threshold) or low (below a given threshold). This provides a feedback loop from the EBNeuro EEG device: when the high level of a synchronization signal appears along with the EEG data, it means, these data belong to the informant’s reaction to the stimulus. To avoid switching to the next word before the reaction to the previous one is fully recorded, the “Word Stimulus” operator is locked by the synchronization signal.

“Write EEG” operator stores the EEG data to the file of a standard EDF format. There are two instances of this operator in the DFD. The upper one stores the informant’s reaction to each word in an individual file. The data are buffered, whenever the “Write” input of the operator receives “True”. When it is changed from “True” to “False” (which means, the word is no longer presented), the file is written to the disk. The name of the file is concatenated from the values of “File Number” and “Filename” inputs of the “Write EEG” operator, prefixed with the informant’s code that is set up via the operator’s settings. If the file with the generated name already exists, the name is suffixed with the number.

The lower “Write EEG” operator stores the EEG data of the informant when no stimulus is presented. Instead of the word, “String Constant” is supplied as a filename, defining a common part for all the names of corresponding files.

For presenting image-based and auditory stimuli, the DFD looks the same, but the operator “Word Stimulus” is replaced by the “Image Stimulus” and “Audio Stimulus” correspondingly. In case, a new type of stimuli is needed, the required operator can be added to SciVi by extending the SciVi ontology, without modifying the source code of the platform.

In the current version of SciVi tools, both visual and auditory stimuli are presented using the PyGame Python library (<https://www.pygame.org/>). Right now, words (provided as a list), images (provided as PNG and JPG files), and sounds (provided as WAV, MP3, and OGG files) can be presented as stimuli, which covers all the preliminary experiments we have carried out for now. However, the SciVi platform already contains appropriate rendering tools to perform more complex scientific visualization on single-board microcomputers like Orange Pi and Raspberry Pi [15], so they can be used whenever needed.

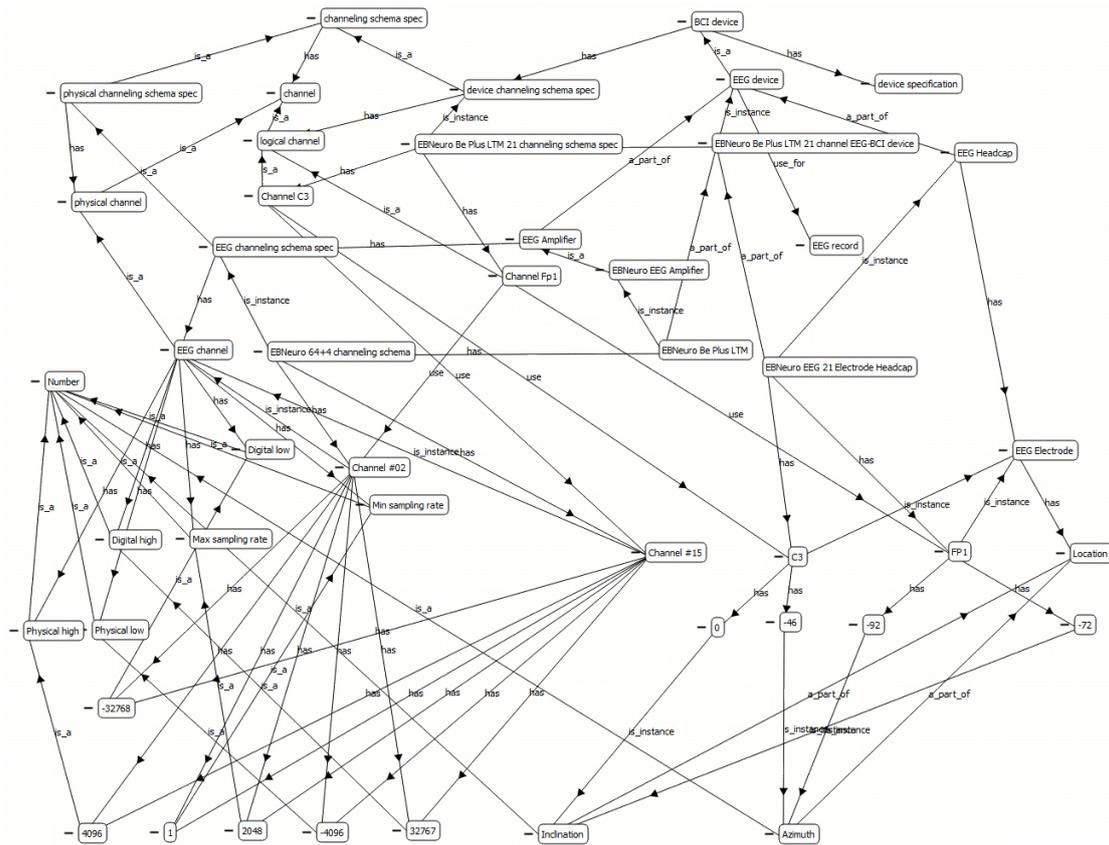
Besides declaring stimuli representation, DFDs can be used to compose and process EEG data processing pipelines as well. For this, corresponding operators are needed, which allow the experiment director to set up required transformations for both real time obtained and pre-recorded data. Lightweight processing operators can be executed on the server- or client side, while the operators involving complex calculations are automatically moved to the SciVi Processing Node (see Fig. 1) that is hosted on the powerful PC.

### 3.3. Ontology-Driven Pipeline Processing

We demonstrate a unified approach to ontology-driven processing of audio-visual stimuli representation pipeline by an example of experiments within the project “Conceptualization of Social Reality in Mass Communication: Cognitive Information Modeling Using Machine Learning Methods, Visual Analytics and Neurocognitive Technologies” (State Assignment No. FSNF-2020-0023, Research Project of Perm State University, 2020–2022). At the current stage of our research, we focus on finding the EEG patterns of the reaction to different concepts (words and texts) displayed on the screen or played back with a speaker.

We subclassed “EEG Device” from BCI-O [16] to use the description of our hardware “EBNeuro Be Plus LTM” with a 16-electrode headcap as ontology-driven mechanism. BCI-O ontology describes generic scenarios of BCI-environment interaction, as well as common properties of EEG-based BCIs. We propose using a lightweight ontology, which model contains two sets: the thesaurus and the set of basic relations. The thesaurus specifies BCI-related concepts, such as “EEG device”, “EEG channel”, “EEG electrode”, etc. In order to reduce the complexity of the ontology reasoner allowing to embed it to Edge devices as firmware [17], we restricted the set of relation types of BCI-O by the paradigmatic types only, such as “has”, “a\_part\_of”, “use”, “use\_for”, “is\_instance”, and “is\_a”. The fragment of proposed ontology is shown in Fig. 3.

We introduce physical parts of “EEG device” concept: “EEG Amplifier” and “EEG Headcap” that has “EEG Electrodes”, as well as split “channeling schema spec” into physical and logical layers, represented by “physical channel” and “logical channel” concepts respectively. Physical channels represent amplifier ADC inputs having their ADC properties, such as minimum and maximum sampling rates, physical and digital limits, etc. Logical channels tie physical channels



**Figure 3:** Fragment of enriched BCI-O ontology.

to particular electrodes, which have their headcap locations. Using these concepts, we describe our experimental equipment as “EBNeuro Be Plus LTM 21 channel EEG-BCI device” (with a “EBNeuro Be Plus LTM 21 channeling schema spec” of logical channels) that consists of “EBNeuro Be Plus LTM” amplifier (with a “EBNeuro 64+4 channeling schema” of physical channels) and “EBNeuro EEG 21 Electrode Headcap” (with corresponding electrodes). This description model ensures flexible mapping of EEG data processing algorithms between different BCI hardware.

#### 4. Visual Tools Supporting EEG-Based BCI Research

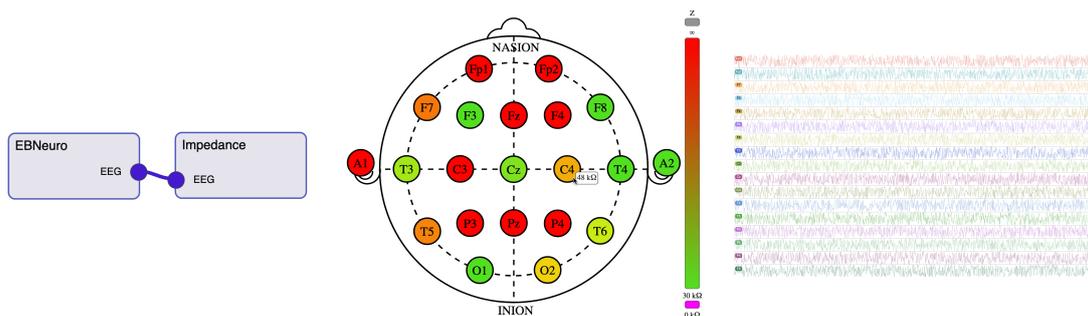
The main goal of the present work is to create a flexible stimuli representation pipeline to conduct experiments involving EEG. However, almost every EEG-based experiment requires a visual inspection of the data being collected. The problem is that the EEG device is very sensitive to electromagnetic noise of different nature, so, at least at the beginning of the experiment, the impact of different noise factors should be reduced as much as possible.

First of all, the contact of the headset electrodes with the scalp skin of the informant should be good enough. According to the advice of experts in neurophysiology, the impedance of

electrodes should be no more than 30 kOhm. To lower the impedance, special conductive gel is used. But the amount of gel that should be applied cannot be defined in advance, it should be picked up experimentally, because it depends on the informant’s hair density, hairstyle, skull shape, etc. It must be noted, that too much gel can short-circuit the neighbor electrodes spoiling the signal. So, there should be a real time monitoring tool to check the impedances of particular headset electrodes to see whether more gel should be applied.

The EBNeuro device has a special mode to measure impedances and transmit them instead of regular potentials. To allow the experiment director to monitor these data, we adopted an SVG image of a standard international 10-20 electrode placement system. Each electrode is painted by the red-to-green color scale according to its impedance. Hovering the electrode pictogram by the mouse cursor opens a pop-up with an actual impedance value. The corresponding DFD diagram and visualization results in SciVi (for the 21-electrode headset) are shown in Fig. 4 (left, middle).

Next, after the headset is ready, the EEG signal should be inspected, whether it contains significant noise. For this, a line chart of potentials and a histogram of frequencies should be visualized for each EEG channel. We implemented a minimalist WebGL-based charting engine that enables very fast visualization suitable for real time EEG signal monitoring. This engine is available as an “EEG Chart” visualization operator that is used in the DFD shown in Fig. 2. The result of signal rendering in SciVi is shown in Fig. 4 (right).



**Figure 4:** SciVi DFD declaring a visual inspection of impedances (left), visualization of impedances (middle), and visualization of the EEG signal in SciVi (right).

Although the visualization methods mentioned above are quite traditional, they are essential for conducting EEG-based experiments, so they should be included in any corresponding data processing pipeline. The SciVi visual analytics platform contains a lot of visual analytics tools [1, 15] organized according to the principles of cognitive graphics [18]. In the future, we plan to adopt them for performing more complex visual analytics of EEG data. However, this requires corresponding data processing mechanisms to be implemented (including the machine-learning-based clustering, classification, etc.), which we are working on.

## 5. Conclusion

In this paper, we propose new SciVi capabilities for flexible and configurable hardware-software pipeline to represent auditory and visual stimuli for conducting EEG-based experiments. The

core of this pipeline is the ontology-driven visual analytics platform SciVi that allows declaring the data obtaining, transformation, storing, and visualization steps through the high-level graphical programming language based on DFDs. Two levels of configurability are achieved. First, the experiment director can combine required data processing operators to suit the conditions of a particular experiment. Second, the knowledge engineer can extend SciVi with new operators describing them in the SciVi ontology, without modifying the platform's source code.

The distinctive feature of the proposed toolset is the automatic distribution of data acquisition, storage, processing, and visualization on different computing nodes in the network, which balances the computation load and allows utilizing various hardware platforms joint with different EEG devices and different stimuli controllers.

The proposed methods and tools created have been used for EEG-based studying of the people's reactions to the displayed news materials related to popular media events. The most significant mass-media concepts are extracted (like "power", "court", "democracy", "opposition" etc.), whereby the aim of the future research is to compare verbal and non-verbal (measured by EEG) reactions of informants to these concepts. The hypothesis is that the verbal reaction does not always match the actual emotions caused by the stimulus, since the verbal reactions can be affected by stereotypes and other external factors.

The high-level graphical user interface of SciVi allows the application domain specialists without advanced IT skills to conduct EEG-based experiments involving complex data transformations and visualization.

## Acknowledgments

The reported study is supported by the Ministry of Science and Higher Education of the Russian Federation, State Assignment No. FSNF-2020-0023 (Research Project of Perm State University, 2020–2022).

## References

- [1] K. Ryabinin, K. Belousov, S. Chuprina, Novel Circular Graph Capabilities for Comprehensive Visual Analytics of Interconnected Data in Digital Humanities, *Scientific Visualization* 12 (2020) 56–70. doi:10.26583/sv.12.4.06.
- [2] M. Rashid, N. Sulaiman, A. P. P. Abdul Majeed, R. M. Musa, A. F. Ab. Nasir, B. S. Bari, S. Khatun, Current Status, Challenges, and Possible Solutions of EEG-Based Brain-Computer Interface: A Comprehensive Review, *Frontiers in Neurorobotics* 14 (2020). doi:10.3389/fnbot.2020.00025.
- [3] A. Schlögl, C. Brunner, BioSig: A Free and Open Source Software Library for BCI Research, *Computer* 41 (2008) 44–50. doi:10.1109/MC.2008.407.
- [4] G. Schalk, D. McFarland, T. Hinterberger, N. Birbaumer, J. Wolpaw, BCI2000: a General-Purpose Brain-Computer Interface (BCI) System, *IEEE Transactions on Biomedical Engineering* 51 (2004) 1034–1043. doi:10.1109/TBME.2004.827072.

- [5] Y. Renard, F. Lotte, G. Gibert, M. Congedo, E. Maby, V. Delannoy, O. Bertrand, A. Lécuyer, OpenViBE: An Open-Source Software Platform to Design, Test, and Use Brain–Computer Interfaces in Real and Virtual Environments, *Presence* 19 (2010) 35–53. doi:10.1162/pres.19.1.35.
- [6] C. A. Kothe, S. Makeig, BCILAB: a Platform for Brain–Computer Interface Development, *Journal of Neural Engineering* 10 (2013). doi:10.1088/1741-2560/10/5/056014.
- [7] R. Oostenveld, P. Fries, E. Maris, J.-M. Schoffelen, FieldTrip: Open Source Software for Advanced Analysis of MEG, EEG, and Invasive Electrophysiological Data, *Computational Intelligence and Neuroscience* 2011 (2010). doi:10.1155/2011/156869.
- [8] I. P. Susila, S. Kanoh, K.-i. Miyamoto, T. Yoshinobu, xBCI: A Generic Platform for Development of an Online BCI System, *IEEJ Transactions on Electrical and Electronic Engineering* 5 (2010) 467–473. doi:10.1002/tee.20560.
- [9] B. Venthur, S. Scholler, J. Williamson, S. Dähne, M. Treder, M. Kramarek, K.-R. Müller, B. Blankertz, Pyff—A Pythonic Framework for Feedback Applications and Stimulus Presentation in Neuroscience, *Frontiers in Neuroscience* 4 (2010) 179. doi:10.3389/fnins.2010.00179.
- [10] L. Richard, D. Charbonneau, An introduction to E-Prime, *Tutorials in Quantitative Methods for Psychology* 5 (2009) 68–76. doi:10.20982/tqmp.05.2.p068.
- [11] K. I. Forster, J. C. Forster, DMDX: A Windows Display Program with Millisecond Accuracy, *Behavior Research Methods, Instruments, & Computers* 35 (2003) 116–124. doi:10.3758/BF03195503.
- [12] J. W. Peirce, PsychoPy – Psychophysics software in Python, *Journal of Neuroscience Methods* 162 (2007) 8–13. doi:10.1016/j.jneumeth.2006.11.017.
- [13] J. Joshi, S. Saharan, P. K. Mandal, BOLDsync: a MATLAB-Based Toolbox for Synchronized Stimulus Presentation in Functional MRI., *Journal of neuroscience methods* 223 (2014) 123–32. doi:10.1016/j.jneumeth.2013.12.002.
- [14] D. H. Brainard, The Psychophysics Toolbox, *Spatial Vision* 10 (1997) 433–436. doi:10.1163/156856897X00357.
- [15] K. Ryabinin, M. Kolesnik, Automated Creation of Cyber-Physical Museum Exhibits Using a Scientific Visualization System on a Chip, *Programming and Computer Software* 47 (2021) 161–166. doi:10.1134/S0361768821030099.
- [16] S. José, R. Méndez, Modeling Actuators in BCI-O: A Context-Based Integration of SOSA and IoT-O, in: *Proceedings of the 8th International Conference on the Internet of Things*, 2018. doi:10.1145/3277593.3277914.
- [17] K. Ryabinin, S. Chuprina, Ontology-Driven Edge Computing, *Lecture Notes in Computer Science* 12143 (2020) 312–325. doi:10.1007/978-3-030-50436-6\_23.
- [18] Y. I. Nechaev, A. B. Degtyarev, A. V. Boukhanovsky, Cognitive computer graphics for information interpretation in real time intelligence systems, in: *Computational Science – ICCS 2002*, 2002, pp. 683–692. doi:10.1007/3-540-46043-8\_69.