

SZCZEGÓŁOWY OPIS PRZEDMIOTU ZAMÓWIENIA (OPIS TECHNICZNY SYSTEMU INFORMATYCZNEGO)

Przedmiotem zamówienia jest wykonanie i wdrożenie oprogramowania, jego integracja z portalem eeg.pl a także dostawa serwera przeznaczonego do obsługi wykonanego oprogramowania, zgodnie ze szczegółowym poniższym opisem.

Wykonawca powinien też dostarczyć platformę sprzętową (serwer klasy PC) pozwalającą na zainstalowanie oprogramowania będącego przedmiotem zamówienia. Opis wymagań zawarty jest w załączniku nr 2 do Specyfikacji Istotnych Warunków Zamówienia.

Zamawiający nie dopuszcza możliwości składania ofert częściowych. Pod uwagę brane będą wyłącznie oferty obejmujące całość zamówienia.

Zamawiający nie dopuszcza możliwość składania ofert wariantowych (odmienny niż określony przez Zamawiającego sposób wykonania zamówienia publicznego nie będzie brany pod uwagę).

Warunki gwarancji dla oprogramowania są rozumiane jako zobowiązanie Wykonawcy do usuwania błędów w dostarczonym oprogramowaniu, rozumianych jako niezgodność działania dostarczonego systemu ze specyfikacją z Załącznika 1, wykazanych w sposób powszechnie przyjęty dla raportowania błędów oprogramowania, w terminie adekwatnym do skali wymaganych korekt, jednak w żadnym przypadku nie dłuższym niż 1 miesiąc.

Dostarczone oprogramowanie musi być objęty pełną gwarancją sprzętową na okres minimum 3 miesięcy (liczony od daty dostawy przedmiotu zamówienia).

Opis techniczny przedmiotu zamówienia

Przedmiotem zamówienia jest wykonanie systemu informatycznego od analizy sygnału EEG, system powinien być zbudowany jest na podstawie istniejącego kodu (SignalML, mpview, Artefakt, Stager). System powinien być z serwera obliczeniowego udostępnionego na portalu eeg.pl, apletu do prezentacji danych i prostych analiz (MP, MMP) dostępnego na stronach portalu

Definicje

SignalML – język opisu struktury danych binarnych (w szczególności wielokanałowych zapisów EEG), przedstawiony w [1], zdefiniowany bliżej na <http://eeg.pl/SignalML>.

MP, MMP – algorytm analizy sygnałów (odpowiednio jednokanałowych i wielokanałowych), zaimplementowany w języku C przez Zamawiającego. Poprzednie implementacje opisane są m.in. w [2] i dostępne na <http://eeg.pl/mp>.

Artefakt – system detekcji fragmentów wielokanałowych zapisów EEG snu (polisomnogramy), opracowany przez Zamawiającego w języku Matlab, opisany m.in. w [3].

Stager – system oznaczania stadiów snu w wielokanałowych zapisach EEG snu (por. np. [4]), opracowywany aktualnie przez Zamawiającego w postaci prototypów w języku Matlab.

MPview – interaktywne oprogramowanie prezentujące wyniki dekompozycji sygnału algorytmem MP, zapisane na dysku w wyspecyfikowanym przez Zamawiającego formacie, m.in. w postaci map gęstości energii. Aktualna implementacja w języku Java dostępna jest pod adresem <http://eeg.pl/mp>, w ramach zamówienia przewidziane jest stworzenie od podstaw odpowiedniego kodu z rozszerzoną funkcjonalnością, m.in. pracy w formie plugina.

Specyfikacje modułów oprogramowania

1 Serwer obliczeniowy i węzeł obliczeniowy

Zadaniem rozproszonego serwera obliczeniowego jest realizacja funkcji obliczeniowych zleczanych przez oprogramowanie klienckie do połączonych siecią komputerową węzłów obliczeniowych.

1.1 Główne funkcje serwera obliczeniowego:

- generacja klas obsługi formatów danych zapisanych w SignalML oraz realizacja tzw. wirtualnej przeglądarki sygnałów na potrzeby portalu eeg.pl
- wyznaczenie rozkładu jednocanałowego sygnału algorytmem MP
- wyznaczenie rozkładu wielocanałowego sygnału algorytmem MMP
- wyznaczenie rozkładu czynności elektrycznej mózgu przy pomocy określonego algorytmu odwrotnego (np. model dipolowy) na podstawie rozkładów wielocanałowych MMP.
- Funkcje specyficzne dla modułów Stager i Artefact opisanych dalej
- kolejki oraz rozproszenie zadań obliczeniowych
- konsola administracyjna pozwalająca na monitorowanie i sterowanie zasobami serwera.

1.2 Architektura serwera obliczeniowego

Serwer powinien się składać z następujących modułów:

- dowolnej liczby węzłów obliczeniowych
- jednego węzła zarządzającego, który będzie odpowiedzialny za sprawowanie kontroli nad obliczeniami wykonywanymi w węzłach obliczeniowych, Węzeł zarządzający stanowi jedyny punkt, do którego odwołują się klienci serwera.
- brokera komunikatów JMS odpowiedzialnego za przekazywanie informacji pomiędzy różnymi węzłami oraz pomiędzy klientami a węzłem zarządzającym
- servletu do zarządzania serwerem obliczeniowym

1.3 Metoda obliczeniowa

Powinna zostać stworzona klasa (lub klasy) reprezentująca dowolną metodę obliczeniową (do wykorzystania w roli singletonu). Klasa ta powinna zawierać i obsługiwać :

- globalnie unikalny identyfikator metody
- obsługę rozpoznawalnej nazwy/identyfikatora metody oraz jej wersji
- możliwość określenia wymaganych danych wejściowych, wymaganej konfiguracji oraz postaci generowanych wyników
- ramowy algorytm wykonywania zadania

W konkretnych podklasach konieczna będzie implementacja:

- głównej metody obliczeniowej działającej na dostarczonym modelu danych
- (opcjonalnie) dodatkowej metody walidującej dostarczone dane i konfigurację
- (opcjonalnie) metody inicjalizującej algorytm przed pierwszym użyciem i ew. metody zamykającej algorytm

1.4 Węzeł obliczeniowy

Węzeł obliczeniowy określony jest poprzez następujące parametry:

- zaimplementowane i aktywne metody obliczeniowe
- wielkość puli równoległych wątków obliczeniowych
- dane “adresowe” pozwalające na połączenie się z węzłem zarządzającym

Węzeł przyjmuje zlecenia obliczeniowe i kieruje je do dowolnego wolnego wątku obliczeniowego. Węzeł obliczeniowy obsługuje następujące komunikaty serwera zarządzającego:

- żądanie wykonania obliczeń
- żądanie udzielenia informacji o podstawowym stanie
- żądanie udzielenia informacji o zadaniu
- żądanie zaniechania wykonywania zadania
- żądanie zmiany parametrów węzła t.j. aktywność danego algorytmu czy liczba dostępnych wątków (w przypadku zmniejszenia działające wątki najpierw kończą obliczenia, a dopiero potem znikają)
- żądanie zamknięcia węzła (bieżące operacje są kończone albo przerywane)

2 Bezpieczeństwo systemu

Projekt zakłada, że jedynym punktem dostępu klientów do serwera będzie węzeł zarządzający.

Komunikacja z klientem oraz wewnętrzna powinna odbywać się z wykorzystaniem protokołu SSL.

Dodatkowo powinny zostać zaimplementowane mechanizmy uwierzytelniania bazujące na: (1) liście użytkowników / haseł oraz (2) uwierzytelnienia wspólnego z uwierzytelnieniem na portalu eeg.pl

Dostęp do konsoli zarządzającej powinien być zabezpieczony odrębnym mechanizmem konto/hasło.

3 Moduły obliczeniowe

3.1 Moduł obliczeniowy – MP5

Istniejąca aplikacja napisana w języku C zostanie dostosowana do pracy z systemem Tworzony system będzie ją wywoływał jako zewnętrzny proces z komunikacją poprzez system plików i standardowe wejście / wyjście.

Aplikacja MP będzie wywoływana przez system dla dobrze określonych niepodzielnych fragmentów sygnału. W przypadku przeliczania dla całego snu system będzie po kolei zlecał obliczenia dla kolejnych składek.

Możliwe będzie przerwanie danego uruchomienia MP, ale nie będzie możliwe zapisanie stanu niezakończonych obliczeń dla danej składki.

Zamawiający zmodyfikuje istniejącą aplikację MP w ten sposób, żeby miała ona możliwość raportowania postępów obliczeń.

3.2 Moduły obliczeniowe – Artefact i Stager

Ogólne funkcjonalności modułu Artefakt:

- automatyczne oznaczanie artefaktów w sygnale EEG
- detaliczne porównywanie różnych oznaczeń artefaktów dla tego samego zapisu EEG (podczas przeglądania sygnału wraz z oznaczeniami)
- zbiorcze statystyczne porównywanie różnych oznaczeń artefaktów dla tego samego zapisu EEG wraz z generacją krzywych ROC i obliczaniem parametrów skuteczności detekcji (AUC, graniczna oraz optymalna czułość i specyficzność detektora, itp.)
- zapis uzyskiwanych oznaczeń w zadanym formacie wynikowym
- zapis do plików wyników pośrednich parametryzacji artefaktów do wykorzystania w innych analizach sygnału EEG

Ogólne funkcjonalności modułu Stager:

- automatyczne wyznaczanie hipnogramu - oznaczanie stadiów snu - w sygnale EEG bez dodatkowej informacji o artefaktach

- automatyczne wyznaczanie hipnogramu - oznaczanie stadiów snu - w sygnale EEG z wykorzystaniem oznaczenia artefaktów (zewnętrznego lub wygenerowanego w ramach pluginu SignalML-Artefakt)
- oznaczanie cykli snu (REM/NREM)
- detaliczne porównywanie różnych hipnogramów dla tego samego zapisu EEG (podczas przeglądania sygnału wraz z oznaczeniami)
- zbiorcze statystyczne porównywanie różnych hipnogramów dla tego samego zapisu EEG wraz z generacją krzywych ROC (multiroc) i obliczaniem parametrów skuteczności detekcji (AUC, graniczna oraz optymalna czułość i specyficzność detektora, itp.)
- zapis uzyskiwanych oznaczeń w zadanym formacie wynikowym
- zapis do plików wyników pośrednich oznaczania stadiów snu do wykorzystania w innych analizach sygnału EEG

Metody obliczeniowe zawarte w istniejącym module MatLab powinny zostać przepisane na język Java. Dopuszczalne jest użycie systemów generacji kodu takich jak: "Matlab Builder for Java". Sposób przekazywania danych i zlecenia procesów obliczeniowych będzie identyczny jak dla MP5

4 Aplikacja kliencka

4.1 Ogólne założenia

Aplikacja powinna zostać rozszerzona o nowe funkcje i przeprojektowana pod kątem standaryzacji interfejsu użytkownika.

Powinna zostać dodana możliwość lokalnego wykonywania obliczeń numerycznych oraz możliwość komunikacji z serwerem obliczeniowym i zdalnego wykonywania obliczeń dla metody MP5. Obliczenia nie będą mogły być kontynuowane po zamknięciu aplikacji, a wszelkie pośrednie wyniki będą w tym wypadku tracone.

Niektóre obliczenia będą mogły zostać wstrzymane (niekoniecznie natychmiastowo, ale jak najszybciej) i ponownie wznowione po ponownym uruchomieniu aplikacji .

Okno aplikacji zostanie podzielone standardowo na menu, toolbar, obszar roboczy oraz status bar. Obszar roboczy okna podzielony zostanie na trzy części (najpierw poziomo, potem część górna pionowo). Stosunki podziału będą konfigurowalne przez użytkownika. W ten sposób utworzone zostaną 3 sekcje: lewa, prawa i dolna. Możliwa będzie łatwa maksymalizacja obszaru sekcji prawej.

Prezentacja samego sygnału powinna zostać zrealizowana podobnie do wersji obecnie istniejącej. Nad sygnałem znajdzie się lokalny toolbar zawierający dostęp do funkcji takich jak antyaliasing czy ustawienia skali, oraz montaż.

Konieczne będą zmiany pozwalające na akomodację wielu tagów dla jednego sygnału:

- należy zaniechać wyświetlania tagów stronicowych na tle sygnału. Do wyświetlania tagów stronicowych przewidziany zostanie specjalny pasek poniżej sygnału, który będzie też zawierał podziałkę czasu.
- tagi blokowe i kanałowe powinny być wyświetlane jak dotąd na tle sygnału, odpowiednio jako pasy pionowe i poziome.

W przypadku jednoczesnego otwarcia więcej niż jednego pliku taga dla wszystkich typów oznaczeń stosowana powinna być zasada podziału przestrzeni kolorowanej w kierunku pionowym na tyle pasów ile tagów jest w danej chwili otwartych i pokazywanych. Tagi stronicowe powinny być pokazywane jedno pod drugim w swoim pasku, tagi blokowe powinny rozciągać się w pionie na wysokość wynikającą z podziału, tagi kanałowe powinny być podzielone podobnie jak tagi stronicowe. Dodatkowo należy wprowadzić możliwość wprowadzenia widoku w tryb porównywania dwóch wybranych tagów, w którym to trybie specjalnie wyróżniane będą różnice.

Proponuje się też zmianę organizacji oznaczania przez upodobnienie jej do ogólnie dostępnych programów graficznych. W osobnej liście rozwijanej wystąpią dostępne w aktywnym tagu typy oznaczeń. Po wybraniu typu oznaczeń użytkownik będzie mógł "malować" tym oznaczeniem po sygnale. Dodatkowo z listy tej dostępne będą trzy narzędzia do zaznaczania fragmentów sygnału (stronicowe, blokowe, kanałowe) oraz "rączka" - alternatywna metoda przewijania sygnału.

Widok sygnału powinien też umożliwić łatwe precyzyjne zaznaczanie fragmentów sygnału (nie tylko przez zaznaczanie myszą).

Dla zaznaczonego fragmentu sygnału możliwe będzie wywoływanie metod obliczeniowych. Metody obliczeniowe uruchamiane będą w tle (lokalnie bądź na serwerze obliczeniowym) i uwidaczniane w sekcji dolnej wraz ze zbiorczym miernikiem postępu. Możliwe będzie otwarcie dodatkowego niemodalnego okna dla każdego zadania, w którym prezentowana będzie jego szersza charakterystyka oraz dokładniejsze mierniki postępu. Po zakończeniu obliczeń wyniki będą opcjonalnie otwierane w sekcji prawej (w zależności od typu metody albo jako nowy zestaw tagów do otwartego sygnału, albo jako nowa zakładka).

Wywołanie metody obliczeniowej będzie poprzedzone ekranem konfiguracyjnym metody. Na ekranie tym dostępne będą wszystkie parametry dopuszczone do konfiguracji przez użytkownika. Każdy taki ekran będzie zawierał też możliwość wyboru zestawu ustawień spośród zapisanych w aplikacji (presetów), możliwość zapisu obecnych ustawień jako zestawu zapisanego oraz analogicznie z zapisem i odczytem pliku XML z dowolnego miejsca na dysku. Dodatkowo dla metod obliczeniowych MP dostępna będzie funkcja zapisu ustawień do pliku konfiguracyjnego MP. System będzie zawsze automatycznie pamiętał ostatnio wybrane ustawienia, nawet po zamknięciu aplikacji.

4.2 Mapy Wignera

Widok **mapy Wignera** będzie odpowiadał jednej otwartej książce. Podobnie jak w obecnej aplikacji mpview pod mapą dostępna będzie pełna rekonstrukcja sygnału oraz rekonstrukcja wybranych atomów. Możliwa będzie nawigacja po książce w sensie wyboru strony i kanału (z przyciskami poprzedni/następny i tak dalej), oraz ustawianie poziomu powiększenia wraz

z narzędziem lupy znanym z narzędzi graficznych. Powiększenie będzie jednocześnie ograniczało widoczny zakres częstości i czasu, w związku z czym nie będzie konieczne (ani możliwe) przewijanie widoku mapy. Wykresy rekonstrukcji będą dostosowane do wybranego poprzez powiększenie zakresu czasu.

Dodatkowo aplikacja będzie zawierała następujące funkcje:

- w osobnym oknie będzie można obejrzeć tabelę atomów składających się na dany rozkład.
- będzie możliwość próbkowania wartości wykresu (czas, częstość, gęstość energii w punkcie) przy pomocy myszy
- będzie możliwość wyboru spośród 4 skal gęstości energii (liniowa, logarytmiczna, pierwiastkowa oraz 'contrast' jak w obecnej implementacji mpview)
- będzie możliwość wyboru pomiędzy wyświetlaniem w skali szarości a wyświetlaniem w kolorach tęczy
- będzie możliwość filtrowania atomów przez podanie przedziału wartości danego parametru, przez dołączenie warunków na występowanie wybranych oznaczeń we wskazanym przez użytkownika tagu lub przez wskazanie dowolnej klasy realizującej filtr atomów na podstawie ich parametrów wg. ustalonego interfejsu. Opcjonalnie aplikacja może udostępniać wbudowany edytor takich klas z możliwością dynamicznej kompilacji.
- będzie możliwość uśrednienia map składających się na książkę (opcjonalnie z zastosowaniem obecnie wybranego filtra)
- będzie możliwość wskazania sygnału oryginalnego dla danego rozkładu spośród obecnie otwartych w aplikacji (i do tego ewentualnie wskazania kanału). Wykres sygnału będzie pojawiał się nad rekonstrukcją.
- będzie możliwość zapisu mapy w prostym formacie tekstowym pozwalającym na względnie proste odtworzenie danych w Matlabie
- opcjonalnie będzie możliwość zapisu mapy w postaci obrazka gif
- z w.w. widoku będzie też dostępna funkcja "atomy na tagi" z opcją zapisu do pliku na dysku lub "dosypania" oznaczeń do wybranego otwartego pliku tagu

4.3 Pozostałe sekcje i widoki aplikacji

Interfejs aplikacji powinien zawierać też następujące elementy:

Widok wyniku iteracyjnego stosowania algorytmów - Widok otwierany po zakończeniu stosowania algorytmu Artefakt lub Stager w trybie iteracyjnym. Będzie zawierał wykres krzywej ROC oraz szereg wartości w postaci arkusza właściwości (w tym warunki początkowe oraz AUC).

Widok przestrzeni roboczej - Widok ten sprowadza się do listy otwartych i ostatnio otwieranych dokumentów z podziałem na typy (sygnały, książki, tagi). Prezentuje też zakładki z sekcji prawej, które są otwarte ale nie posiadają konkretnej reprezentacji w systemie plików.

Widok otwartych sygnałów - W tym widoku pokazane są w postaci drzewa wszystkie otwarte sygnały. Po rozwinięciu sygnału pojawia się lista stron sygnału.

Widok otwartych książek - W tym widoku pokazane są w postaci drzewa wszystkie otwarte książki.

Widok otwartych tagów - W tym widoku pokazane są w postaci drzewa otwarte pliki tagów i oznaczenia w nich zawarte.

Widok właściwości - Widok właściwości pokazuje standardowy arkusz właściwości dla wybranego elementu w sekcji lewej (o ile wybrany element posiada właściwości)

Sekcja lewa - W sekcji lewej wyświetlane będą informacje w postaci drzew. Przewidziane są następujące zakładki:

- przestrzeń robocza - łatwy dostęp do otwartych i ostatnio otwieranych plików sygnałów oraz książek, a także dostęp do listy pozostałych otwartych obecnie danych tymczasowych takich jak policzone ad hoc i niezapisane książki oraz wyniki iteracyjnego stosowania algorytmów
- otwarte sygnały
- otwarte książki
- otwarte tagi (pogrupowane wg. sygnałów)

Sekcja dolna - W sekcji dolnej wyświetlana jest kolejna grupa zakładek:

- konsola - miejsce na komunikaty aplikacji
- aktywne zadania obliczeniowe - w tabeli pokazywane są aktualnie wykonywane zadania obliczeniowe wraz z ich miernikiem postępu.
- właściwości - arkusz właściwości obiektu podświetlonego w sekcji lewej. W przypadku wybrania atomu pokazywane są parametry atomu, w przypadku taga parametry taga, w przypadku sygnału parametry sygnału i tak dalej.

Widok zadań obliczeniowych - Zakładka ta pokazuje tabele zleconych zadań obliczeniowych. Dla każdego zadania pokazywany jest typ, czas zlecenia, informacja o przedmiocie zlecenia, status oraz miernik postępu.

Konsola - Konsola pokazuje komunikaty aplikacji.

4.4 Konfiguracja

Konfiguracja aplikacji powinna odbywać się poprzez standardowy dialog z wieloma zakładkami.

4.5 Instalator aplikacji

Aplikacja powinna zostać wyposażona w multiplatformowy (Windows, Linux) łatwy w użyciu instalator, przypominający standardowe instalatory często spotykane przez użytkowników systemu MS Windows. Liczba czynności wymaganych od użytkownika przed rozpoczęciem pracy powinna być jak najmniejsza.

5. Applet demonstracyjny

Powinien zostać stworzony applet Java zawierający silnie okrojoną funkcjonalność pełnej aplikacji klienckiej opisanej wyżej. Applet powinien być osadzony na stronach portalu eeg.pl. Funkcje appletu:

- ładowanie i przeglądanie sygnałów oraz nałożonych na nie tagów, z tym że możliwe będzie otwarcie tylko jednego taga na sygnał
- modyfikowanie oznaczeń, ale bez wsparcia dla przeglądania drzewa tagów, tagi będą widoczne tylko na tle sygnału, a ich parametry będzie można obejrzeć w modalnym dialogu zawierającym arkusz właściwości
- zapis zmodyfikowanych oznaczeń
- konfiguracje i uruchamianie metod obliczeniowych na zaznaczonych kawałkach sygnału. Wszelkie obliczenia prowadzone będą na serwerze obliczeniowym. Applet nie będzie wspierał obliczeń lokalnych ani nie będzie wspierał prowadzenia wielu obliczeń w tle. Jednocześnie możliwe będzie wykonywanie tylko jednego zadania obliczeniowego, którego postęp będzie prezentowany w modalnym okienku analogicznym do okienka zadania widocznego w aplikacji. Możliwe będzie jedynie zlecenie zaniechania wykonywania zadania (żadne wstrzymywanie ani wznawianie nie będzie wspierane). Jediną metodą możliwą do uruchomienia w aplocie będzie metoda MP5.
- ładowanie książek i przeglądanie map Wignera (w tym książek wygenerowanych przez metodę MP uruchomioną z poziomu appletu). Nie będzie natomiast wspierał przeglądania książek w postaci drzewa (panel lewy w aplikacji). Dla danej mapy Wignera będzie można obejrzeć tabelę atomów (w osobnym oknie lub w osobnej zakładce). Dostępna będzie też funkcja filtrowania atomów, ale tylko wg. prostych przedziałów parametrów (bez obsługi filtrowania po tagu i filtrowania klasą dostarczoną przez użytkownika).
- zapis wygenerowanych książek

6. Literatura

[1] SignalML: metaformat for description of biomedical time series P.J. Durka and D. Ircha, Computer Methods and Programs in Biomedicine Volume 76, Issue 3, pp. 253-259, December 2004

Nr postępowania: WF-37-43/07

SZCZEGÓŁOWY OPIS PRZEDMIOTU ZAMÓWIENIA

Załącznik nr 1

[2] Stochastic time-frequency dictionaries for Matching Pursuit, P.J. Durka, D. Ircha and K.J. Blinowska. IEEE Transactions on Signal Processing, vol. 49, No. 3, pp. 507-510, March 2001.

[3] A simple system for detection of EEG artifacts in polysomnographic recordings P. J. Durka, H. Klekowicz, K. J. Blinowska, W. Szelenberger, Sz. Niemcewicz, IEEE Transactions on Biomedical Engineering, April 2003, Vol. 50 No 4, pp. 526-528

[4] High resolution parametric description of slow wave sleep Piotr J. Durka, Urszula Malinowska, Waldemar Szelenberger, Andrzej Wakarow, Katarzyna J. Blinowska, Journal of Neuroscience Methods vol. 147/1 pp. 15-21, 2005