SOFTWARE REQUIREMENTS SPECIFICATION (v1.0)

PROJECT:

Visualization of the Human Cognition Using Brain Data

Group Name: KernelPanic

Group Members:

Esen Aytan	1819036
Zeynep Büşra Çınar	1819804
Irmak Doğan	1819242
Kamyar Ghasemlou	1786896

PREFACE

This document contains the software requirements specification for the “Visualization of Human Cognition Using Brain Data” software. The document is prepared according to the “830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.”

This Software Requirements Specification provides a complete description of all the software requirements and views of the “Visualization of Human Cognition Using Brain Data”.

The first section of this document includes scope and purpose of project and gives overall description of it. The following sections include detailed description and requirements of the project.

REVISION HISTORY

Version	Date	Changed	A/D/M	Brief Description
1.0	30/11/2014	-	-	Initial Version
2.0	03/01/2015	2.2.11, 2.2.12, 2.2.13, 2.2.14, 2.2.15	A	New use cases are added.

*A: Added, M: Modified, D: Deleted

CONTENTS

1. INTRODUCTION

This document is a software requirement specification for Visualization of the Human Cognition Using Brain Data project. Its aim is giving information about the whole of the document and making the table of contents one step further. In this document, firstly we are going to define the problem and introduce the purpose and the scope of this document. Secondly, we are going to give an overall description. After these steps, we are going to state specific requirements, data models and behavioral models with their descriptions consecutively. Finally, we are going to present our planning, team structure, basic schedule and process model.

1.1 PROBLEM DEFINITION

Developing ways to study mental representations has been a long-standing and intriguing challenge to computer scientists. MVPA is a very recent approach that few researchers have implemented in neuroimaging research. MVPA uses machine learning algorithms to train on fMRI data, and can be used to classify the type of information being processed or cognitive operation that is being carried out, based on distributed patterns of neural activity in the brain. In its current state, MVPA is a very advanced and promising approach that can enable researchers to answer questions about cognitive operations and brain function that could not be assessed with other approaches. However, given its recent nature, the algorithms researchers use for MVPA has much room to improve. In this project, we propose an interdisciplinary collaboration that aims to fill this gap in the literature. Specifically, we propose several fMRI studies, whereby the feasibility of the current machine learning algorithms in the literature can be tested, and the algorithm with maximum performance can be identified. With this approach, via seeking out the machine learning approach that can best describe neural activity in the brain associated with cognitive processes, our aim is to build a model, called Local Voxel Networks that can be used to classify neural activity assessed by fMRI during cognitive operations. The suggested approach has three main steps:

Step 1: Acquiring the fMRI data from experiments

Step 2: Enhancing the fMRI data by image processing techniques to prepare for MVPA techniques

Step 3: Developing a new model for fMRI image and comparing it to the available techniques

Above, there is the definition of "Tubitak Project 112E315" which is in process at the Image processing and Pattern Recognition Laboratory of Computer Engineering Department, Middle East Technical University. During our senior project, we are going to be part of this project. Since it is a huge project and there is time limitations, our focus is going to be implementing the 2nd step of the aforementioned study only. Also, our project is going to be a follow-up of CEREBRA which was one of the previous year's design projects. They have achieved that the project has showed 3D brain image at a particular time by reading a set of brain data files. Our goal is to develop this project by adding new features and creating 3D brain animation by using different data sets. Instead of data sets which are include data at a point time, we are going to use another data sets which are include data from at a set of time. Thus, we are going to convert the 3D image to 3D animation.

1.2 PURPOSE

The purpose of this document is to give a detailed description of the project. By giving details of the architecture, functionalities and specifications, it will explain the purpose and features of the system that will be developed. This document is going to serve as a guideline for both the development team and the users. Since more than one version of this document are going to be released, there may be modifications to adapt changes of requirements and specifications of the project. The target audience of this project is mostly academicians and medical institutes.

As stated in the project proposal, Visualization of the Human Cognition Using Brain Data is fundamentally important in neuroanatomy, neurodevelopment, cognitive neuroscience and neuropsychology. Neuroscientists can benefit from this tool to evaluate how risky brain surgery or similar treatment is for a patient and to learn how a normal, diseased or injured brain is functioning. They can map the brain with this tool to identify regions linked to critical functions such as speaking, moving, sensing, or planning. This is useful to plan for surgery and radiation therapy of the brain. Clinicians also can benefit from this to anatomically map the brain and detect the effects of tumors, stroke, head and brain injury, or diseases such as Alzheimer's. This is useful to plan for surgery and radiation therapy of the brain. Clinicians also can benefit from this to anatomically map the brain and detect the effects of tumors, stroke, head and brain injury, or diseases such as Alzheimer's.

1.3 SCOPE OF THE PROJECT

Final product of this project is a high-end desktop/compute cluster application, named "Visualization of the Human Cognition Using Brain Data". The software is responsible for analysis and visualization of fMRI data. Cognitive neuroscience is an emerging area or study. These studies aim to understand how human brain actually works. However, brain imaging tools generate data that is generally too big for a human to analyze. The product will be used in cognitive neuroscience studies to present this data in a more understandable manner. Software will first read brain imaging data, and then analyze it to see connections, merge closely connected components and visualize and animate the data via different visualize options so that it is easy to inspect and see connections.

This project will be done by 4 people in 2 semesters.

1.4 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

3D	3 Dimensional
FC	Functional Connectivity
fMRI	Functional Magnetic Resonance Imaging
GPU	Graphics Processing Unit
GUI	Graphical User Interface
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
MATLAB	Matrix Laboratory
METU	Middle East Technical University
MNI	Montreal Neurological Institute
MVPA	Multivoxel Pattern Analysis
RAM	Random Access Memory
SRS	Software Requirements Specification
UML	Unified Modeling Languages
Unity3d	Unity is a cross-platform game creation system developed by Unity Technologies, including a game engine and integrated development environment (IDE).

1.5 REFERENCES

1. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.
2. http://neuro.ceng.metu.edu.tr/
3. http://en.wikipedia.org/wiki/Functional_magnetic_resonance_imaging
4. Star UML Guide, retrieved from http://staruml.sourceforge.net/docs/userguide(en)/toc.html
5. http://en.wikipedia.org/wiki/Lobes_of_the_brain
6. Xia M, Wang J, He Y (2013) BrainNet Viewer: A Network Visualization Tool for Human Brain Connectomics. PLoS ONE 8: e68910

1.6 LITERATURE SURVEY AND EXISTING SOLUTIONS

FMRI: It is a special type of MRI that makes it possible to watch different areas of a person's brain while they do different tasks. A Functional MRI shows the use of oxygen in different brain areas. The Deoxyhemoglobin in blood absorbs a radio wave signal and later emits it (sends it back out). Oxyhemoglobin does not. When an area of the brain is active, the blood flow to that area increases and more oxyhemoglobin is delivered to supply the active cells with oxygen. This means less of the signal is absorbed. An active area of the brain absorbs and emits less energy than a less active area. This can be seen live as it is happening when different areas of the brain 'light up' on the images as they become active.

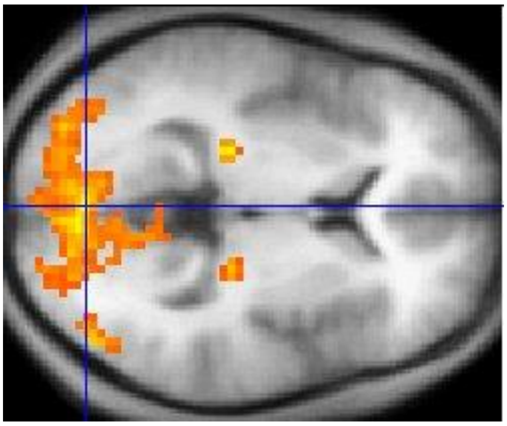


Figure 1 : A Picture from fMRI

BRAIN REGIONS:

- **The frontal lobe** is located at the front of the brain and is associated with reasoning, motor skills, higher level cognition, and expressive language. This area of the brain receives information from various lobes of the brain and utilizes this information to carry out body movements. Damage to the frontal lobe can lead to changes in sexual habits, socialization, and attention as well as increased risk-taking.
- **The parietal lobe** is located in the middle section of the brain and is associated with processing tactile sensory information such as pressure, touch, and pain. A portion of the brain known as the somatosensory cortex is located in this lobe and is essential to the processing of the body's senses. Damage to the parietal lobe can result in problems with verbal memory, an impaired ability to control eye gaze and problems with language.
- **The temporal lobe** is located on the bottom section of the brain. This lobe is also the location of the primary auditory cortex, which is important for interpreting sounds and the language we hear. The hippocampus is also located in the temporal lobe, which is why this portion of

the brain is also heavily associated with the formation of memories. Damage to the temporal lobe can lead to problems with memory, speech perception, and language skills.

- **The occipital lobe** is located at the back portion of the brain and is associated with interpreting visual stimuli and information. The primary visual cortex, which receives and interprets information from the retinas of the eyes, is located in the occipital lobe. Damage to this lobe can cause visual problems such as difficulty recognizing objects, an inability to identify colors, and trouble recognizing words.

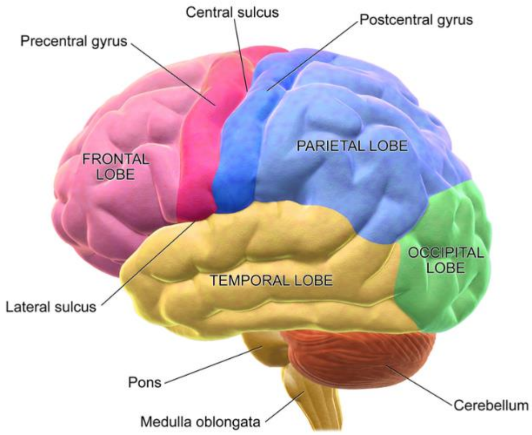


Figure 2: Lateral View of the Brain

Image taken from:

http://en.wikipedia.org/wiki/Lobes_of_the_brain#mediaviewer/File:Blausen_0101_Brain_LateralView.png

BRAIN NET VIEWER:

BrainNet Viewer is a brain network visualization tool, which helps researchers to visualize topological patterns of structural and functional brain networks on different ways.

It has several important features:

- BrainNet Viewer is an open-source, Matlab-based software package with a graphical user interface (GUI), illustrating human connectomes as ball-and-stick models;
- BrainNet Viewer can 1) display brain networks in multi-views; 2) display combinations of brain surface, nodes and edges; 3) adjust properties of network elements (i.e., nodes and edges); 4) map the volume image to brain surface; 5) support various types of image format exporting and video making; 6) provide interactive operations, such as zoom and rotate; and 7) illustrate the interactions between two brain networks.

1.7 OVERVIEW

IEEE std 830-1998 standard is used for this documentation. Overall description partis provide general view and aims to simplify the understanding of Specific Requirements:

- In the first section of the SRS, purpose of project and scope of the project are explained.
- Second section of the SRS is the overall description of the project.
- In the third section of the SRS specific requirements are explained in a detailed way.
- In the fourth and fifth sections, data and behavioral models are given.
- Sixth section of the SRS is the explanation for the planning part of the project.
- In the seventh section of the SRS conclusion is made and project is briefly summarized.

2. OVERALL DESCRIPTION

CEREBRA is a project that shows you brain data in a specified way. Brain voxels can be seen as a nodes and edges and can be filtered to analyze brain structure. Also, brain change in time can be observed with the animation. This project relies on the properties that are generally present in brain analyze projects. In this part of the SRS, we will focus on all determinants, which affects the product and its requirements.

The main topic we covered in this part can be summed up as:

- *Product Perspective*
- *Products Functions*
- *User Characteristics*
- *Constraints*
- *Assumptions and Dependencies*
- *Apportioning of Requirements*

2.1 PRODUCT PERSPECTIVE

This project consist of taking the brain data and manipulating this data to achieve some meaningful results. At the beginning we have a brain data but and deduction cannot be done with this data because of some reasons such as it is not in the correct coordinate system or it doesn’t have brain template. After some operations, this brain data become ready to make deductions. These operations can be done via a GUI which is provided by unity ports. Moreover, this project enables user to see brain changes in terms of time as an animation.

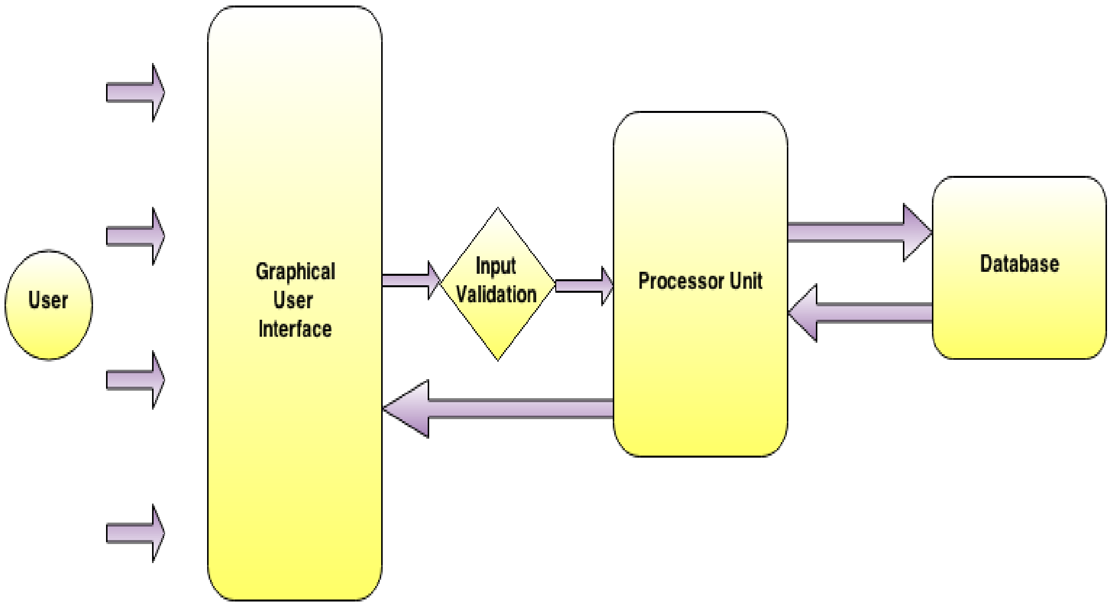


Figure 3: Block Diagram of the System

2.1.1 SYSTEM INTERFACES

The system is already read brain scan (fMRI) data, we are planning to convert these scanned data to brain regions and analyze this regions. To achieve this with best modularity, we plan to decouple these subsystems. So there will be an input subsystem, an analysis subsystem and a visualization subsystem; all of them connected via well-defined data interfaces.

2.1.2 USER INTERFACES

GUI is provided in the project. There are buttons and selection layers on the screen so that user can filter the brain data to make it meaningful. Required filters can be chosen by user thanks to this GUI.

Function names in the GUI gives user idea about what this function makes. Also, in the GUI functions have some messages for user to make the correct selection. For example if user tries to make voxel suppression this functions warns the user about the minimum and maximum values in the intensity values. The default language is English in this application.

2.1.3 HARDWARE INTERFACES

The process needed for visualizing the human brain data is computationally expensive. The project needs a compute cluster or a high end processor. Again, memory demand is high. Visualization part also needs a Unity capable graphics card. Also, some calculations may be side loaded to GPU, so a good graphics card will benefit the project.

2.1.4 SOFTWARE INTERFACES

This project requires Unity and Visual Studio to work. Unity 4.5.5 version should be used and Visual Professional 2013 version should be provided for this project. Also it needs Windows as an operating system for now, but it can be extended in the future. Windows7 and Windows8 is appropriate for this project. Moreover, C# is used while implementation the project. This are all, closed source but some of trailer versions can be used.

Purpose of this software interfaces is that these software tools makes our project well oriented.. Also it enables the user work effectively.

2.1.5 COMMUNICATION INTERFACES

Communication is done by unity ports. Moreover buttons are provided for the user to increase the communication between the user and application. To reach this application, all project codes should be downloaded and build.

2.1.6 MEMORY

The project uses input data in the range of gigabytes. This data has to be read once and converted to a more compact format, maybe compressed in the process.

Assuming 80,000 nodes and a 80,000£80,000 adjacency list, the software needs at least 80,000x80,000xsizeof (float) memory for one frame of data. That is 25,600,000,000 bytes for the best case, nearly 24 gigabytes of continuous data.

For secondary storage purposes, the project has much more challenging requirements. A typical dataset contains 30 or more frames. So, a single dataset takes up 720 gigabytes of data with an uncompressed file format.

2.1.7 OPERATIONS

In User Interfaces section (2.1.2), most of the operations are explained, so it will not be mentioned again here.

2.1.8 SITE ADAPTATION REQUIREMENTS

Brain data should be provided for this application as an input. And size of brain data can be change in terms of which region it belong to or what operations and filters will be applied. No safety limit is obligatory for this brain data.

Moreover, user should supply appropriate brain data for their purpose. For instance, if they want to tag a brain region, this region should be provided in the input file.

2.2 PRODUCTS FUNCTIONS

There is no grouping for the user. All users have same power in this application.

2.2.0 COMMON USE CASE

Use case of all functionalities in the application is showed below:

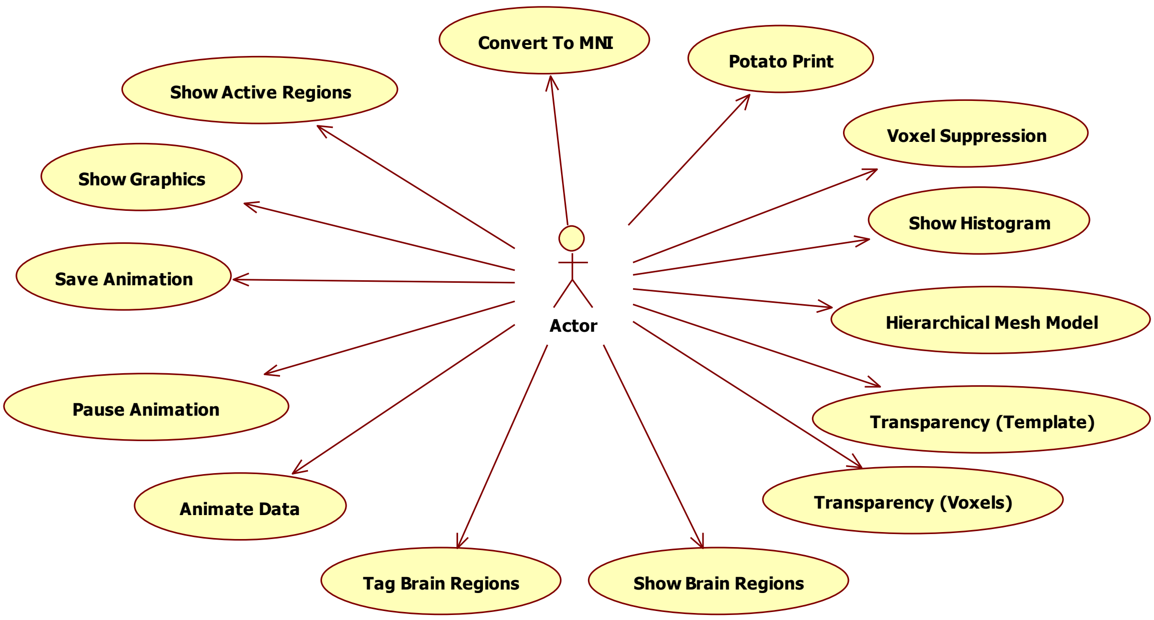
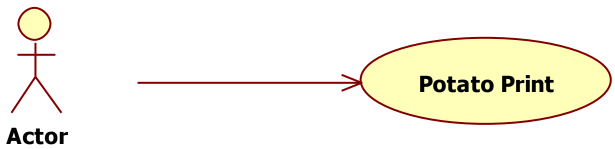


Figure 4: Use Case Diagram of the Project

These functions will be explained in upcoming sections in detail.

2.2.1 POTATO PRINT USE CASE

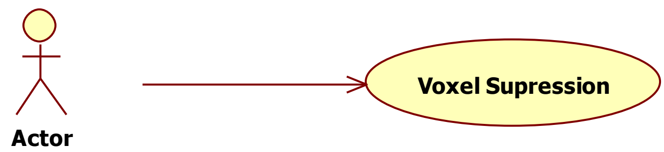


Description: Selecting and printing a voxel slice as a potato slice.

Detailed Description:

- After selecting this function, user should determine the axis which slices will be taken.
- After selection of axis, all slices in that axis will be shown in a way that window will be divided portions and each partition will show each slice.
- If user click one slice, this slice will be shown bigger so that user can examine this slice in detail.
- Thanks to this function, user will have a chance to look each slice separately and analyze characteristic of each slice in brain data.

2.2.2 VOXEL SUPPRESSION USE CASE

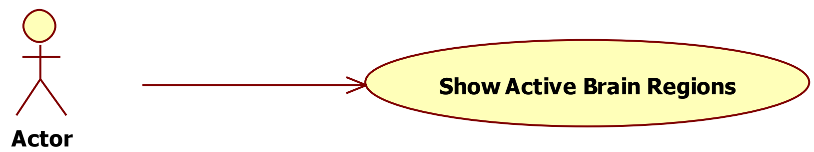


Description: Selecting the voxel which has some intensity value.

Detailed Description:

- After selecting this function, user should determine between which intensity value range he/she wants to see.
- This function warns you about maximum and minimum numbers of the intensities.
- After selecting the range of the intensity, application will show user the voxel which is between this intensity values.

2.2.3 SHOW ACTIVE BRAIN REGIONS USE CASE

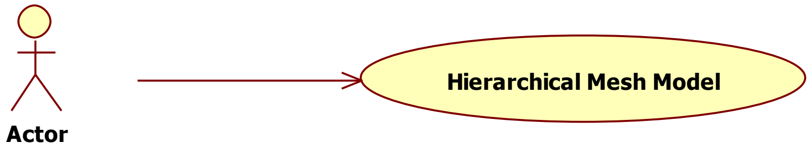


Description: Showing active regions of the brain.

Detailed Description:

- After pressing this button, system scans region with intensity change is too much.
- Brain regions with stable intensity changes are hidden from the brain presentation.
- Thanks to this function, user can observe the activity percentage of a brain region.

2.2.4 HIERARCHICAL MESH MODEL USE CASE

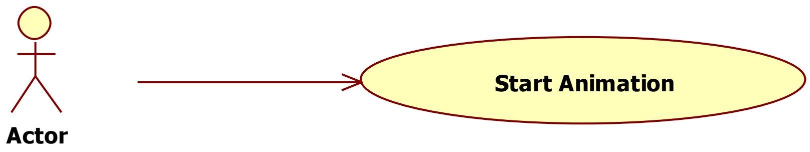


Description: Showing brain meshes model change according to hierarchical level.

Detailed Description:

- After pressing “Up Level” button, System takes the average of the intensity of the regions have close intensity. The number of regions decreases. Fewer amounts of regions are shown.
- If user continues to press that button, regions with the close intensity merge and shown updated regions.
- If user presses “Down Level” button, brain regions will return previous level.
- If user continues to press that button, regions with the close intensity separate and brain gets closer to initial situation.

2.2.5 START ANIMATION USE CASE

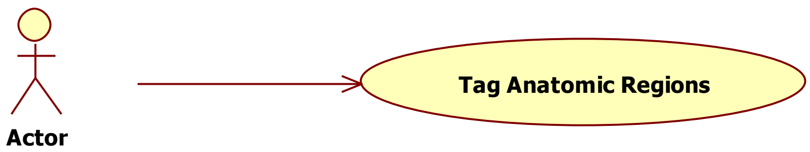


Description: Showing change of the brain voxels in terms of time.

Detailed Description:

- After selecting this function, animation will start on the screen.
- In this animation, change in the voxels in term of time will be shown to the user.
- This animation, enables user to observe activity percentage of the regions in each time frame.
- Thanks to this function, user can observe each region in each time frame.

2.2.6 TAG ANATOMIC REGIONS USE CASE

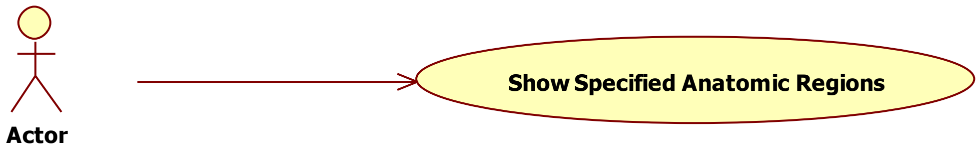


Description: Showing brain' anatomic regions in different colors and naming on tooltip screen.

Detailed Description:

- After selecting this function, system will give different color to anatomic regions to show detailed differences.
- Users' mouse should hover on brain regions. If a mouse stop on a region, tooltip screen will be shown. In tooltip screen, name of anatomic region will be shown to users.
- Thanks to this function, user can tag the anatomic regions on the brain.

2.2.7 SHOW SPECIFIED ANATOMIC REGIONS

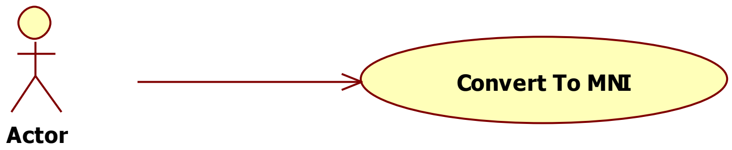


Description: Showing only selected brain regions on the screen.

Detailed Description:

- After selecting this function, system will give different color to anatomic regions to show detailed differences. If user ticks the checkbox to select a region, they will see the information about this region.
- After tick process is done, system will hide unselected regions.
- Thanks to this function, users can see specified anatomic region on the screen.

2.2.8 CONVERT TO MNI USE CASE

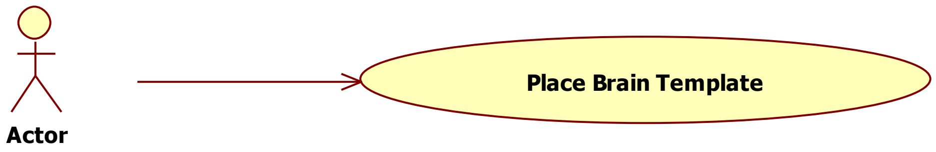


Description: Showing Brain structure in the MNI coordinate system.

Detailed Description:

- After pressing “MNI Coordinates” button, System will convert data space according to “MNI Coordinates” with transition matrix.
- Brain voxels will pass through the MNI coordinate space.
- Thanks to this function, users can show “real” brain anatomic structure. On the screen.

2.2.9 PLACE BRAIN TEMPLATE USE CASE

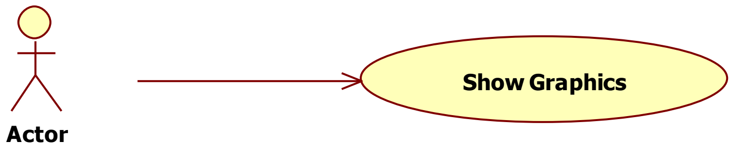


Description: Showing brain template model to the users.

Detailed Description:

- After pressing "Brain Template" button, System will find brain template model and it will add brain template surroundings of voxels, which are, converted MNI coordinate before.
- After this selection, brain template model will be shown on the screen.
- Thanks to this function, users can see brain with the actual brain template.

2.2.10 SHOW GRAPHICS USE CASE

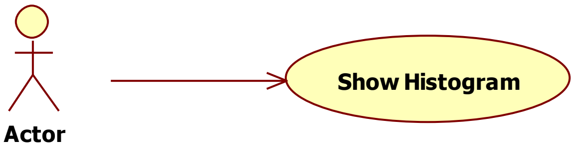


Description: In a time interval, showing intensity value of a brain region in a graphic.

Detailed Description:

- While animation process, if user presses the button "Show Graphics.", data used for animation will be shown in a way that window will be divided portions according to pressed time.
- Each situation of animation can be observed in specific time.
- If user clicks on the each graphics, clicked part of a graphic will be shown bigger and more detailed form.
- Thanks to this function, intensity value change can be observed in a specified time interval.

2.2.11 SHOW HISTOGRAM USE CASE

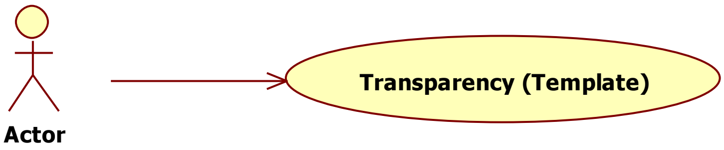


Description: During the voxel suppression process, histogram of intensity values of voxels will be shown.

Detailed Description:

- While voxel suppression process, if user presses the button "Show Histogram".
- The system takes the intensity values and sends a MATLAB script.
- Thanks to this script, the histogram is created and shown in the screen.

2.2.12 TRANSPARENCY ADJUSTMENT (TEMPLATE)

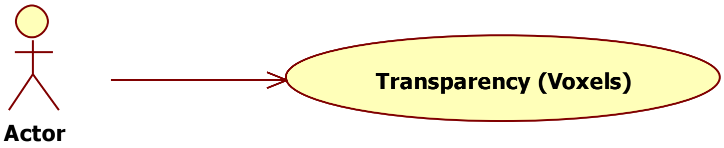


Description: After converting to the MNI, brain template will be added to the brain image, and its' transparency will be adjustable.

Detailed Description:

- The brain template is compatible with the shape of brain in the MNI coordinates.
- There will be a slider to adjust the transparency.
- Thanks to this slider, transparency will be able to decrease and increase.

2.2.13 TRANSPARENCY ADJUSTMENT (VOXELS)

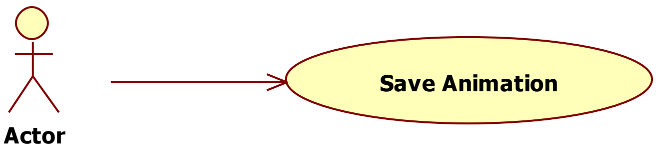


Description: To be able to see inner voxels, transparency of the outer voxels can be adjusted.

Detailed Description:

- There will be a slider to adjust voxel transparency of the outer voxels
- When the transparency increased, the inner voxels will be seen easily.

2.2.14 SAVE ANIMATION

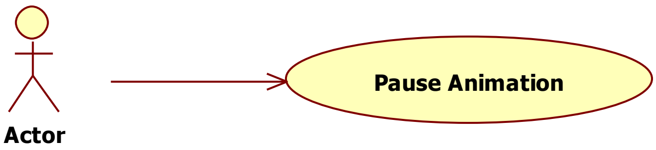


Description: Before the animation process, the user may want to save animation process.

Detailed Description:

- There will be a "Save Animation" button.
- If the user clicks the button, when animation started, animation process will be saved.
- When process is finished, the .mp4 file will be saved automatically to the project folder.

2.2.15 PAUSE ANIMATION



Description: During the animation process, the user may want to pause animation process.

Detailed Description:

- There will be a "Pause Animation" button.
- If the user clicks the button, the animation will pause.
- If the user clicks "Animate Data" button, the animation will resume.

2.3 USER CHARACTERISTICS

There is no difference between user characteristics such as admin or sign up member. However target users are researchers and scientists.

2.4 CONSTRAINTS

First and foremost C++ is the defacto language for developing performance critic applications. However, we choose to implement our program in C#. This is mostly due to our selection of 3D engine and extensibility of the program.

As stated before, the project is being developed for research purposes. Project team queried several researchers about their setup to find most suitable configuration. It was concluded that the project should be developed for Windows platform. Quality advantage of Windows graphics card drivers aided this choice.

Apart from stated two constraints, there are no other constraints for this project. However since this project is scientific, its conclusions are crucial and its safety and security conditions should be satisfied.

2.5 ASSUMPTIONS AND DEPENDENCIES

First of all, the project assumes that the host computer will have enough RAM to load a single frame and enough secondary storage capacity to store whole file in a single disk. Also, it is assumed that host computer (or the client) will have a graphics card with Unity capabilities. There are open-source projects on similar fields of research. They have a great potential for code reuse. However, most of these projects are written in Python or MATLAB languages. Consequentially, they use MATLAB file formats.

The project is expected to be mostly self-contained, using libraries that already exist in the development environment client uses. Project team will try to minimize external dependencies. Unity will definitively be used. For the aforementioned reason, the project will also depend on a library for MATLAB interoperability.

2.6 APPORTIONING OF REQUIREMENTS

These requirements may be delayed now. Up future version of the software, they will be applied. System is conducive to adding new features.

3. SPECIFIC REQUIREMENTS

Requirements can be divided into 3 parts as Interface Requirements, Functional Requirements and Non-Functional Requirements.

3.1 INTERFACE REQUIREMENTS

In our project, we use Unity3d and Microsoft Visual Studio to create GUI. To handle process management we will use a Selection Lists and Buttons. Loading file is also going to be via buttons and windows. For some operations, we will add check boxes. We will add new keyboard short-cuts to make project more usable.

3.2 FUNCTIONAL REQUIREMENTS

In this part, use cases will be explained in detail.

3.2.1 POTATO PRINT

Use Case Name	Potato Print
Xref	Section 2.2.1
Actor	All Users
Trigger	The users enter X or Y or Z coordinates values to label.
Precondition	None.
Basic Path enter information	1) The users decide which slide region of the brain they want to show. 2) According that, Users enter the system X or Y or Z-axis values and slice numbers to field. 3) The system checks unfilled fields. 4) System gets from data voxels' ranks and checks the input. 5) System removes all areas of brain except axis of slice number. 6) Users can see the specific slice of brain they want to show. 7) All slices in that axis will be shown in a way that window will be divided portions and each partition will show each slice.

	8) If users click one slice, this slice will be shown on big screen.
Alternative Path	None
Post Condition	Slice presentation of the whole brain.
Exception Paths	None
Other	None

3.2.2 VOXEL SUPPRESSION

Use Case Name	Voxel Suppression
Xref	Section 2.2.2
Actor	All Users
Trigger	The users enter density range to two labels.
Precondition	None.
Basic Path enter information	1) The users decide to see region which intensity. 2) According that, Users enter the lower limit and upper limit of intensity to input field. 3) The system checks unfilled fields. 4) System gets voxels' having appropriate intensity from data. 5) System removes all areas of brain except range of given input. 6) Users can see the specific area of brain they want to show.
Alternative Path	None
Post Condition	Change in presentation of the whole brain.
Exception Paths	None
Other	None

3.2.3 SHOW ACTIVE REGIONS

Use Case Name	Show Active Regions
Xref	Section 2.2.3
Actor	All Users
Trigger	The users press the “Active Regions” button.
Precondition	None.
Basic Path enter information	1) Users press the button called “Active Regions”. 2) System scans region with intensity change is too much. 3) Brain regions with stable intensity changes are hidden from the brain presentation.
Alternative Path	None
Post Condition	Specific brain regions are only shown.
Exception Paths	None
Other	None

3.2.4 HIERARCHICAL MESH MODEL

Use Case Name	Hierarchical mesh model
Xref	Section 2.2.4
Actor	All Users
Trigger	The users press the “Up Level” or “Down Level” button.
Precondition	None.
Basic Path enter information	1) Users press the button called “Up Level” button. 2) System takes the average of the intensity of the regions have close intensity. 3) The number of regions decreases. Fewer amounts of regions are shown. 4) While users press the “Up Level” button again and again, regions with the close intensity merge and shown updated regions.

	5) Users press the button called “Down Level”. 6) According that, brain regions can return previous level. 7) While users press the “Down Level” button again and again, regions with the close intensity separate and brain gets first situation.
Alternative Path	None
Post Condition	Brain meshes model change according to hierarchical level.
Exception Paths	When brain was in first condition, Down Level doesn’t work.
Other	None

3.2.5 START ANIMATION

Use Case Name	Start Animation
Xref	Section 2.2.5
Actor	All Users
Trigger	The users press the “Start Animation”.
Precondition	None.
Basic Path enter information	1) Users press the button called “Start Animation” button. 2) System starts the animation of the change of the brain voxels. 3) Change in the voxels’ intensity is shown to user in terms of time. 4) User can observe activity percentage of the regions in each time frame.
Alternative Path	None
Post Condition	Activity percentage of brain in each time is shown to users dynamically.
Exception Paths	None
Other	None

3.2.6 TAG ANATOMIC REGIONS

Use Case Name	Tag Anatomic Regions
Xref	Section 2.2.6
Actor	All Users
Trigger	The users run the processor.
Precondition	None.
Basic Path enter information	1) Users run the "Tag Anatomic Regions" processor. 2) System gives different color to anatomic regions to show detailed differences. 3) Users' mouse hover on brain regions. 4) When a mouse stop on an region, tooltip screen is shown 5) In tooltip screen, name of anatomic region is shown to users.
Alternative Path	None
Post Condition	Brain' anatomic regions are shown in different colors and they names are shown in tooltip screen.
Exception Paths	None
Other	None

3.2.7 SHOW SPECIFIED ANATOMIC REGIONS

Use Case Name	Show Specified Anatomic Regions
Xref	Section 2.2.7
Actor	All Users
Trigger	The users run the processor.
Precondition	None.
Basic Path enter information	1) Users run the "Show Specified Anatomic Regions" processor.

	2) System gives different color to anatomic regions to show detailed differences. 3) Users tick the checkbox for select the regions they want to see. 4) After tick process is done, system hidden unselected region from the presentation of brain.
Alternative Path	None
Post Condition	Brain' anatomic regions are shown in different colors and only shown selected regions on the screen.
Exception Paths	None
Other	None

3.2.8 CONVERT TO MNI COORDINATES

Use Case Name	Convert To MNI Coordinates
Xref	Section 2.2.8
Actor	All Users
Trigger	The users press the "MNI Coordinates" button.
Precondition	None.
Basic Path enter information	1) Users press the "MNI Coordinates" button. 2) Systems convert data space according to "MNI Coordinates" with transition matrix. 3) Brain voxels pass from different space MNI coordinate space. 4) Users can show "real" brain anatomic structure.
Alternative Path	None
Post Condition	Brain' anatomic regions are shown in "real" brain structure.
Exception Paths	None
Other	None

3.2.9 PLACE BRAIN TEMPLATE

Use Case Name	Place Brain Template
Xref	Section 2.2.9
Actor	All Users
Trigger	The users press the “Brain Template” button.
Precondition	The users press the “MNI Coordinates” button.
Basic Path enter information	1) Users press the “Brain Template” button. 2) System finds brain template model. 3) System adds brain template surroundings of voxels, which are, converted MNI coordinate before. 4) In the screen, brain template model are shown. 5) Users can do process over the brain template view.
Alternative Path	None
Post Condition	Instead of voxels view, brain template model is present to users.
Exception Paths	System gives warning if user tries to add template to voxels without convert model to MNI Coordinates.
Other	None

3.2.10 SHOW GRAPHICS

Use Case Name	Show Graphics
Xref	Section 2.2.10
Actor	All Users
Trigger	The users run the process of “Show Graphics”.
Precondition	The users started the animation of brain.
Basic Path enter information	1) Users start to animate the brain 2) Users press the button called “Show Graphics.” 3) In the screen, data used for animation is shown in a way that window will be divided portions according to pressed time.

	4) Users can look each situation of animation in specific time. 5) Users click on the each graphics and clicked graphic is shown as bigger and detailed.
Alternative Path	None
Post Condition	Data shown in animation is shown on the screen according to wanted time.
Exception Paths	None
Other	None

3.3 NON-FUNCTIONAL REQUIREMENTS

3.3.1 PERFORMANCE REQUIREMENTS

- Since our system is a single user program, it does not have to be enable multiple users.
- There is no need for spending too much time for understanding.
- Response time should be minimum as much as possible.
- Other response times (Brain Region Separation) should be minimum as much as possible.
- The user will be able to show multiple number of images for a related configuration.
- The user will be able to gets the existing data from the database in at most 2 second.
- The user can capture the image of the animation in less than 1 second.
- The software should give the output in a format that users can show brain in details.

3.3.2 DESIGN CONSTRAINTS

During the development of the project, the fMRI data is going to be provided by Prof. Dr. Fatoş Yarman Vural and during evaluation of the data Brain Net Viewer is going to be used to create a background for this project.

- Final 3D image and animation which will be shown in the graphical user interface is going to be obtained by using Unity3d.
- C# is going to be used as programming language.
- Microsoft Visual Studio 2013 is going to be used as IDE.

- Additionally UML standards will be used for the diagrams and IEEE standards will be used for the reports.

4 DATA MODEL AND DESCRIPTION

4.1 DATA DESCRIPTION

4.1.1 DATA OBJECTS

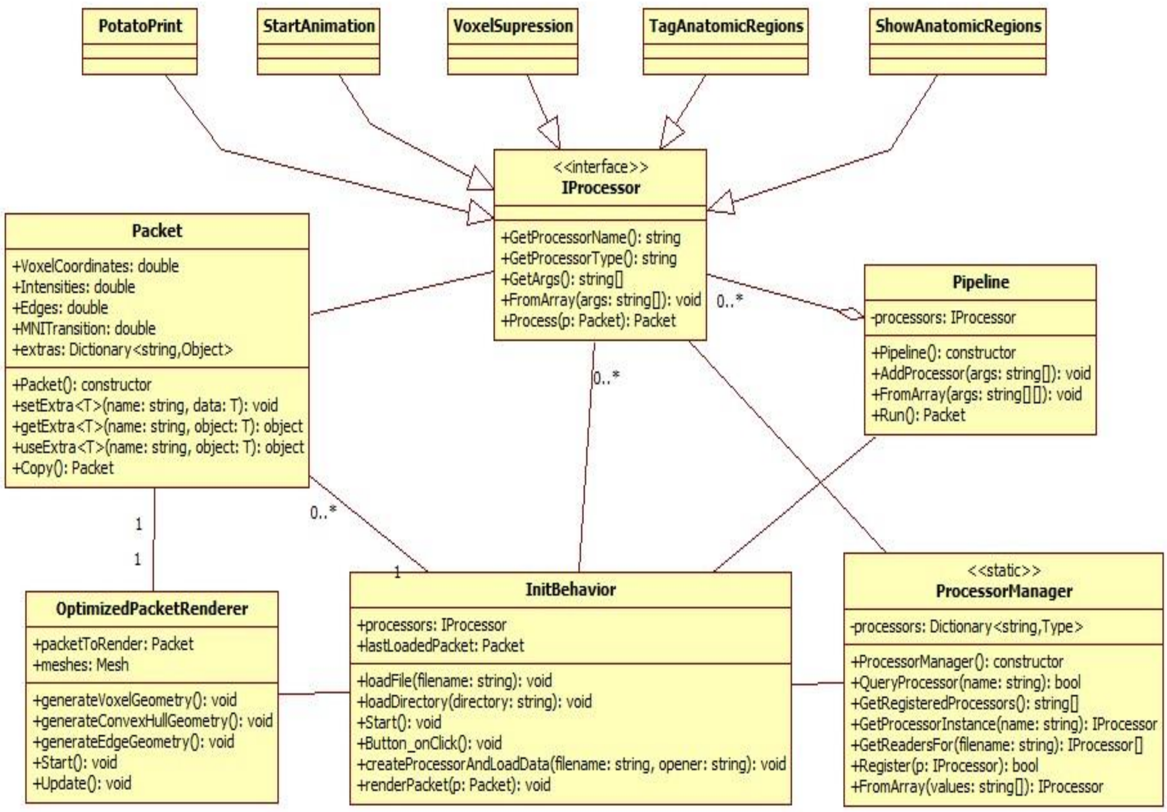


Figure 5: Class Diagram of the Project

PACKET: This class is used as an immediate data format between two Processors. Packet class encapsulates all data needed by Processors: voxel coordinates, edge values, etc... It also offers a way to pass named extra data between Processors.

IPROCESSOR: This interface defines outlines of Processors and how Processors should be implemented.

PIPELINE: This class is responsible for chaining Processor operations. A pipeline is an object that the user can save to or load from it a file. Thus, it also enables the user to create his/her own presets.

When a Processor is added to the Pipeline, Pipeline object checks whether it is the first Processor to be added, and if it is, is it an input type Processor.

PROCESSORMANAGER: This class is responsible for managing Processor selection and generation. This is a static class and its' members are all static. Each Processor must register itself with the ProcessorManager.

INITBEHAVIOUR: This class responsible for merging and starting all operations.

OPTIMIZEDPACKETRENDERER: This class is called from InitBehaviour and it is responsible for rendering the particular packet.

4.1.2 DATA DICTIONARY

Voxels: Nodes to represent set of brain neurons. They are representing as cubes. They have:

- Intensities: represented via different colors
- Size: weight, height of voxels
- Coordinates: places in the screen or brain
- Depth: front or back of screen according to the camera position and voxel coordinates

Edges: Arcs between voxels. They are representing with conical objects. They have:

- Starting Voxel: First point of the arc
- End Voxel : Last point of the arc
- Weights: their lengths that is how much they have neurons.
- Intensities: represented via different colors

5 BEHAVIORAL MODEL AND DESCRIPTION

5.1 DESCRIPTION FOR SOFTWARE BEHAVIOR

The projects workflow is pretty simple. It follows this simple pattern:

1. Read data
2. Process data for visualization and animation
3. Draw data
4. Animate data

When the program opens, it should check whether command line contains any options or input files. If they exist, the program should make necessary changes and/or load the input file. User can cancel the action and close the program at this stage. Program may support loading from different file formats, e.g. MATLAB file format, CSV format or plaintext formats. Readers for all these formats will be written and produce output in the same structure. When users selects a file, program will begin loading the data.

After data is loaded, program will begin processing data. Processed data may be saved for caching purposes, since the processing part will be the most burdensome computation that will be done by the program. In the process, raw data will be compressed by lossy techniques, resulting nodes will be cross checked for any potential correlations. When this process is done, data will be converted to brain coordinates and then, MNI coordinates to make the brain more realistic.

Visualization and animation parts are the next step. Brain coordinates will be converted to physical coordinates and shown on the screen. User can still change settings at this stage to examine the data in more detail. For example, voxel size may be changed (requiring all triangles to be reset), intensity values may be normalized (requiring all triangles to be colored again), and new processes may be applied.

Then, to animate the data, user can load a time series data instead of loading a particular time data. According to the data, the frames on the screen will change and in each step new packet rendering will be done. To speed up the operation, we may examine all voxels and edges between voxels and decide which have changed. Thus, we only change these voxels intensities, size or weights. Instead whole data is rendered from the beginning, we only change those. Finally, we can achieve more smooth and continuous passes between frames.

5.2 STATE TRANSITION DIAGRAM

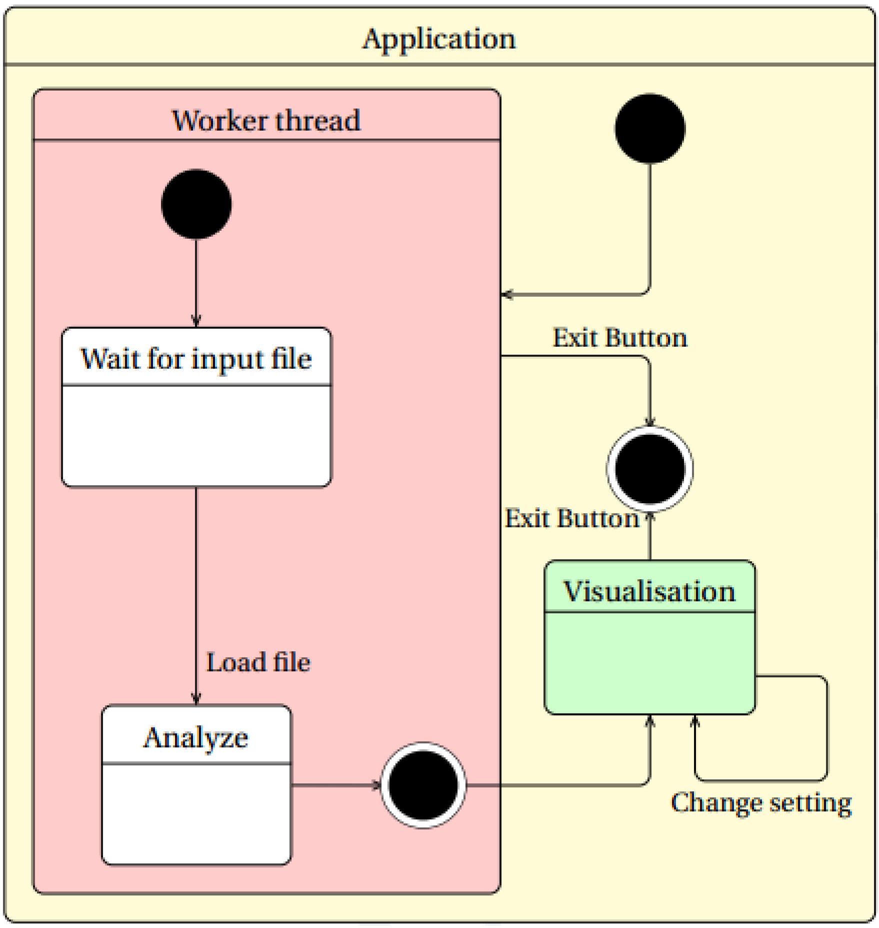


Figure 6: State Transition Diagram of the Project

As shown in the diagram, the application starts with a worker thread and waits an input brain data which is fMRI data. It analyzes the data and bring it in compliance with visualization. During the visualization step, user can do same changes on the data such as normalizing, zooming, converting, rotating, adding new processes etc. A user can exit the program in somewhere of the application.

6 PLANNING

6.1 TEAM STRUCTURE

Prof. Dr. Fatoş Yarman Vural - Project Advisor

Emre Aksan – Project Asistant

Esen Aytan - Researcher, Developer

Zeynep Büşra Çınar - Researcher, Developer

Irmak Doğan - Researcher, Developer

Kamyar Ghasemlou - Researcher, Developer

6.2 ESTIMATION (BASIC SCHEDULE)

The project is planned to be finished by June 2015. There will be 4 revisions. At each revision, the program will become more usable and bug-free. At first revision, it is planned to add some new features to CEREBRA. These are Potato Print Operation, Voxel Suppression. At second revision, we will deal with animation of the brain and converting to MNI coordinates. And also, we try to brain data to place in a brain template. Thus, the brain will become more realistic and comprehensible. Third and fourth revisions will be focused based on client feedback.

6.3 PROCESS MODEL

Project members are going to use Scrum Framework which is Agile Method, which is shown below, for the project. Scrum has three roles, three ceremonies, and three artifacts.

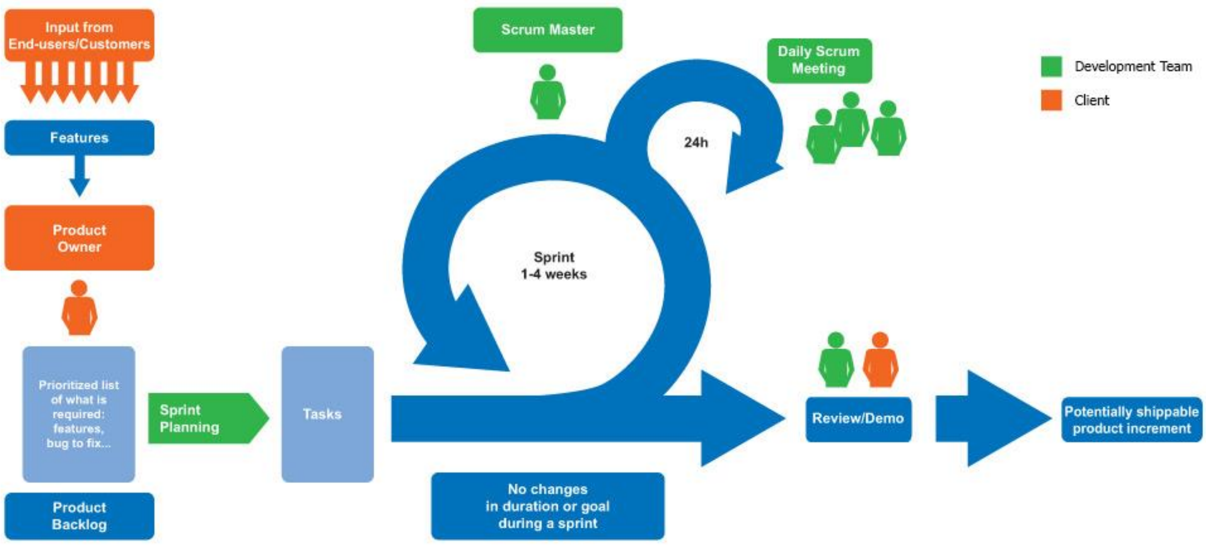


Figure 7: Agile Development using Scrum

Image taken from: http://www.innovateapps.com/process.html

The roles defined by the Scrum framework are Product Owner, Scrum Master and Scrum Team. The Product Owner will have the control of the whole project, and will ensure that the artifacts delivered by the process are in line with the research project aims and the required deliverables. Obviously, Prof. Dr. Fatoş Yarman Vural will have this role. Scrum Master is the person who is responsible for enforcing the rules of the process and ensuring that the process is used as intended. According to the subjects of developing tasks, the scrum master will change in our team. Finally, the Scrum Team is defined as anyone who is involved in the realization of this project.

The three ceremonies are defined as Sprint Planning, Sprint Review, and Daily Scrum Meeting. In the beginning of the sprints, our team will hold a Sprint Planning session, after that, we will work on the requirements of the sprint by supporting them with Scrum Meetings and finally we will review the sprint. Sprints will roughly cover a month; however, Scrum Meetings will not be held daily since all the team members are under tight schedule for their other work.

Finally, the three artifacts are called Product Backlog, Sprint Backlog, and Burndown Chart. The Project Backlog is a prioritized list of project requirements with estimated times to turn them into parts of project deliverables and/or completed system functionality and the Sprint Backlog a list of tasks that defines the team's work for each Sprint. We will record our work activities to the Burndown Charts in order to estimate the time that will be required by the remaining work and to see the whole picture about our project.

7 CONCLUSION

This document gives information about the project "Visualization of the human cognition using brain data" which aims to visualize the brain as 3D image and 3D animation advised by Fatoş Yarman Vural. First of all the aim of the project is described, then our solution approach to this problem is explained in the document. To explain this we have tried to explain basic functionality of the project, interface requirements of the application, performance, attributes, and design constraints imposed on the implementation. In the overall description part, all of the functions that this application is going to be able to do explained. User and function relationships, user roles and characteristics are modeled. The assumptions that will maintain the accuracy are made in order to sustain a reliable application, in addition to that data models and behavioral models are presented. Finally we presented the team structure and basic expected schedule and the process model for the team "Kernel Panic".