INTERFACE-DRIVEN DESIGN: A CASE STUDY IN DEEP BRAIN
STIMULATION DATA MANAGEMENT

by

SISI GU

Submitted in partial fulfillment of the requirements
For the degree of Master of Science

Thesis Advisor: Dr. Guo-Qiang Zhang

Department of Electrical Engineering and Computer Science
CASE WESTERN RESERVE UNIVERSITY

January, 2015
CASE WESTERN RESERVE UNIVERSITY
SCHOOL OF GRADUATE STUDIES

We hereby approve the thesis of

SISI GU

candidate for the Master of Science degree*.

(signed) Dr. Guo-Qiang Zhang, Ph.D.

(chair of the committee)

Dr. Licong Cui, Ph.D.

Dr. Rong Xu, Ph.D.

(date) November 26, 2014

*We also certify that written approval has been obtained for any proprietary material contained therein.
# TABLE OF CONTENTS

| LIST OF TABLES | ................................................................. | v |
| LIST OF FIGURES | ................................................................. | vi |

## 1 Introduction ................................................................. 1

## 2 Background ................................................................. 3

2.1 Parkinson Disease and Its Treatment ........................................... 3
2.2 Deep Brain Stimulation ........................................................... 4
2.3 Movement Disorders Center at UH Neurological Institute ................. 6
2.4 Motivation ................................................................. 8
2.5 Organization of Thesis ......................................................... 10

## 3 DBS-CRADLE™ Software Development ......................................... 11

3.1 Objective ................................................................. 11
3.2 Agile Methodology ............................................................ 13
3.3 Web-Interface-Driven Development ............................................. 15

## 4 The DBS-CRADLE™ DBS System ............................................... 19

4.1 Ruby on Rails ............................................................... 19
4.2 MVC Architectural Pattern .................................................. 20
4.3 System Environment .......................................................... 21
4.4 System Architecture ........................................................ 22
4.5 Database Design ............................................................. 24
4.6 Notable Features of DBS-CRADLE™ ......................................... 25
4.6.1 Interactive Data Validation ................................................. 25
4.6.2 Visual Navigation Aid ..................................................... 29
4.6.3 Expandable and Collapsible Tables ....................................... 29
4.6.4 Synopsis Report View ..................................................... 30
5 **User Interface of DBS-CRADLE™** .................................................. 32

5.1 Authentication ................................................................. 32
5.2 Index Page ................................................................. 34
5.3 Create New Patient ......................................................... 35
5.4 Core-specific Patient Data Management ................................. 37
  5.4.1 Profile ................................................................. 37
  5.4.2 Clinical Visit ......................................................... 38
  5.4.3 Surgery ............................................................... 40
  5.4.4 Programming Review ............................................. 41
  5.4.5 Monopolar Review ................................................. 42
  5.4.6 Adverse Event ...................................................... 43
  5.4.7 Imaging .............................................................. 44

6 **Result** ........................................................................... 46

7 **Conclusion** .................................................................... 48

APPENDIX **Ruby on Rails Code** ........................................... 50

  A.1 Sample Controller ....................................................... 50
  A.2 Sample Model .......................................................... 52
  A.3 Sample View .......................................................... 53

**LIST OF REFERENCES** .................................................... 61
### LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>DBS-CRADLE™ System Environment</td>
<td>22</td>
</tr>
<tr>
<td>5.1</td>
<td>Clinical Visit Types</td>
<td>39</td>
</tr>
<tr>
<td>6.1</td>
<td>Editable Input Fields for Each Editable Interface</td>
<td>47</td>
</tr>
</tbody>
</table>

Appendix Table
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>DBS Clinical Procedures</td>
<td>6</td>
</tr>
<tr>
<td>3.1</td>
<td>Agile Development for DBS-CRADLE™</td>
<td>14</td>
</tr>
<tr>
<td>3.2</td>
<td>WIDD Development Process</td>
<td>16</td>
</tr>
<tr>
<td>4.1</td>
<td>MVC Architectural Pattern</td>
<td>20</td>
</tr>
<tr>
<td>4.2</td>
<td>DBS-CRADLE™ System Architecture</td>
<td>23</td>
</tr>
<tr>
<td>4.3</td>
<td>DBS-CRADLE™ Database Design</td>
<td>24</td>
</tr>
<tr>
<td>4.4</td>
<td>Data Presence Validation Example</td>
<td>26</td>
</tr>
<tr>
<td>4.5</td>
<td>Placeholder and Range Validation Example</td>
<td>26</td>
</tr>
<tr>
<td>4.6</td>
<td>Data Type Validation Example</td>
<td>27</td>
</tr>
<tr>
<td>4.7</td>
<td>Data Uniqueness Validation Example</td>
<td>27</td>
</tr>
<tr>
<td>4.8</td>
<td>Restricted Value Set Example</td>
<td>28</td>
</tr>
<tr>
<td>4.9</td>
<td>Visual Navigation Aid Example</td>
<td>29</td>
</tr>
<tr>
<td>4.10</td>
<td>Expandable and Collapsible Tables</td>
<td>30</td>
</tr>
<tr>
<td>4.11</td>
<td>Partial Synopsis Report for Clinical Visit</td>
<td>31</td>
</tr>
<tr>
<td>5.1</td>
<td>Log in Page: only authorized users can access DBS-CRADLE™</td>
<td>32</td>
</tr>
<tr>
<td>5.2</td>
<td>Sign Up Page: every new user is required to signup in order to gain access</td>
<td>33</td>
</tr>
<tr>
<td>5.3</td>
<td>Index Page: the list of existing patients</td>
<td>34</td>
</tr>
</tbody>
</table>
Appendix

Figure

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4</td>
<td>Creation Page: the main page to create a new patient</td>
<td>36</td>
</tr>
<tr>
<td>5.5</td>
<td>Patient Information Accessing Links</td>
<td>37</td>
</tr>
<tr>
<td>5.6</td>
<td>Patient Profile Page: overview of general patient information</td>
<td>38</td>
</tr>
<tr>
<td>5.7</td>
<td>Clinical Visit Page: overview of all clinical visits</td>
<td>39</td>
</tr>
<tr>
<td>5.8</td>
<td>Surgery Page: all information about DBS surgery</td>
<td>40</td>
</tr>
<tr>
<td>5.9</td>
<td>Programming Page: record programming parameters</td>
<td>41</td>
</tr>
<tr>
<td>5.10</td>
<td>Monopolar Review Page</td>
<td>42</td>
</tr>
<tr>
<td>5.11</td>
<td>Monopolar Review Edit Page</td>
<td>43</td>
</tr>
<tr>
<td>5.12</td>
<td>Adverse Event Page: add and edit adverse effects of DBS</td>
<td>44</td>
</tr>
<tr>
<td>5.13</td>
<td>Imaging Page: CT or MRI scan of the brain</td>
<td>45</td>
</tr>
</tbody>
</table>
List of Acronyms

- DBS: Deep Brain Stimulation
- PD: Parkinson’s Disease
- DBS-CRADLE™: Clinical Research Analytics Data Lifecycle Environment for Deep Brain Stimulation
- WIDD: Web-Interface Driven Design
- UH: University Hospitals
- FDA: Food and Drug Administration
- MOA: Monoamine Oxidase
- MRI: Magnetic Resonance Imaging
- CT: Computed Tomography
- UPDRS: United Parkinson’s Disease Rating Scale
- TRS: Tremor Rating Scale
- BFMDRS: Burke-Fahn-Marsden Dystonia Rating Scale
- TWSTRS: Toronto Western Spasmodic Torticollis Rating Scale
• UDRS: Unified Dystonia Rating Scale

• MMSE: Mini Mental Status Exam

• MoCA: Montreal Cognitive Assessment

• MVC: Model-View-Controller

• RoR: Ruby on Rails

• DRY: Don’t Repeat Yourself

• AC-PC: Anterior Commissure - Posterior Commissure

• IPG: Implantable Pulse Generator

• SF36: Short Form (36) Health Survey
ACKNOWLEDGMENTS

First of all, I would like to express my sincerest gratitude to my adviser and committee chair, Dr. Guo-Qiang Zhang, who has been instructing me since the very beginning of my Master’s program. Prof. Zhang has continuously provided me with numerous constructive and professional academic advice throughout the project as well as my thesis writing.

Next, I would like to thank Shiqiang Tao, who is both a great mentor and lead developer of this project. His guidance, patience and encouragement during the project have made my contribution possible and meaningful.

I would also like to thank Dr. Benjamin Walter, who I have worked closely with for all of his professional consultation in movement disorders and deep brain stimulation.

Furthermore, I would like to send my appreciation to my committee members Dr. Li-cong Cui and Dr. Rong Xu for their genuine and beneficial advice on my thesis.

Last but not least, I’m grateful to Wei Zhu, Yang Chen and all the other members from Center for Clinical Investigation for their help and support.

This research was supported in part by the Case Western Reserve University CTSA Grant NIH/NCATS UL1TR000439.
Interface-Driven Design: A Case Study in Deep Brain Stimulation Data Management

Abstract

by

SISI GU

With Deep Brain Stimulation (DBS) emerging as a powerful therapy for Parkinson’s Disease (PD), the effective capture, retrieval and analysis of data are some of the major informatics challenges. To address the challenges, this dissertation presents the interface-driven design of DBS-CRADLE™ a web-based Clinical Research Analytics Data Lifecycle Environment for Deep Brain Stimulation. DBS-CRADLE™ is designed in a rapid and iterative development cycle by frequently involving domain experts and future users throughout the project lifespan by utilizing an innovative agile development strategy called Web-Interface-Driven Design (WIDD). DBS-CRADLE™ aims to provide (1) data capture interface, providing multiple ways of data entry assistance and validation to ensure the data quality; (2) meaningful data organization, mirroring the workflow of actual patient-clinician interactions; (3) visual navigation aid, facilitating information and web page accessibility. The WIDD design of DBS-CRADLE™ resulted in 38 distinct data capture interfaces and 849
discrete data elements in collaboration with clinical experts at the Movement Disorders Center of the University Hospitals Neurological Institute in Cleveland.
Chapter 1

Introduction

The Movement Disorders Center at the University Hospitals (UH) Neurological Institute in Cleveland offers Deep Brain Stimulation (DBS) which is a treatment for Parkinson’s disease approved by Food and Drug Administration (FDA). In the past two decades, there are approximately 60,000 patients worldwide with PD have undergone this surgery, with an annual accrual of 8000 to 10,000 new patients [1]. Each patient who undergoes that DBS procedure generates huge volumes of data throughout the process. The accuracy, accessibility, legibility and processability of large quantity of patient information and person-specific decision making directly determine the quality of patient care delivery and health outcomes.

The previous system adopted by Movement Disorders Center was for the use of generalized clinical data management, which means the lack of bespoken functionality, customized interface and data management flexibility to be part of the standard operating protocol for DBS. There is no off-the-shelf application found in the market that scientifically documents the electrophysiological data acquired for DBS surgery, permanent electrode implantation
location in the brain, or stimulation variables associated with therapeutic benefit. Furthermore, with potential manufacturers and devices to emerge into the US market, the universal consensus on best practices for the DBS data management system remains undefined.

This dissertation presents DBS-CRADLE™ (Clinical Research Analytics Data Lifecycle Environment for Deep Brain Stimulation) to address the previous issues. DBS-CRADLE™ is a web-based, non-manufacturer dependent data management system for deep brain stimulation. DBS-CRADLE™ is designed to integrate into the clinical workflow and mirror the actual process of DBS procedures. The system aims to capture longitudinal clinical data in three stages: pre-operative registration and assessment, intra-operative surgical data acquisition, and post-operative evaluation and adjustment. DBS-CRADLE™ is able to centralize the data collected from all stages of the procedure, divide the data in a meaningful way, reduces data management complexity, and ultimately improve the efficiency of the entire procedure.
Chapter 2

Background

2.1 Parkinson Disease and Its Treatment

Parkinson’s disease (PD) is one of the most common neurodegenerative disorders. It is usually classified by clinical features such as bradykinesia, tremor and rigidity. According to Parkinson’s Disease Foundation, PD is currently affecting one million people in the United States and an estimated four million worldwide. It is estimated to grow substantially in the next 20 years according to a project conducted in the 5 most populous Western Europe’s nations and the world’s 10 most populous nations. The total number of individuals with PD is expected to be greater than 9 million in 2030 [2].

Medications are available for the treatment of PD such as levodopa, dopamine agonists and monoamine oxidase (MAO)-B inhibitors. These are widely used to alleviate the PD symptoms and reduce the risk of dyskinesia. For the PD patients with major depressive disorder, interpersonal psychotherapy is proved to be a feasible treatment to improve depression symptoms [3]. However, despite the advances of pharmacotherapy and psychotherapy, there are drawbacks to each treatment options. The medications take effect initially, but after five years of treatment a majority of patients no longer benefit from
pharmacotherapy and even develop medication related motor complications [5] [6]. Psychotherapy, a complementary and invasive way [7] which focuses only on depression, is partially or completely ineffective for some patients. Therefore while treatment options are available for most patients, some PD patients are treatment-resistant and even unresponsive to both psychotherapy and pharmacotherapy [4].

2.2 Deep Brain Stimulation

Deep brain stimulation (DBS) is a Food and Drug Administration (FDA) approved neurosurgical treatment [23] for PD patients who cannot benefit from either pharmacotherapy or psychotherapy. It is a clinically proven successful technique that offers treatment for PD with concomitant obsessive compulsive disorder or treatment-resistant depression.

The workflow of DBS procedures is shown in Figure 2.1. First the patient’s demographic and registration information are recorded along with referring physician information. Afterward, multiple clinical visits conducted by the physicians with collaboration of the patient are used to identify the PD rating scales and other medical information which will be used to assess the DBS qualification of the patient. Then, a multidisciplinary team consisting of a movement disorders neurologist, a functional neurosurgeon, and a neuropsychologist conducts a care conference and determines the applicability and optimal surgical plans regarding to the DBS procedure. If qualified, the patient undergo stereotactic imaging procedures such as magnetic resonance imaging (MRI), computed tomography (CT) or ventriculography in order for the physicians to determine the target area in the brain to
implant the microelectrodes prior to the DBS surgery. During the surgery, microelectrodes are precisely implanted in some brain areas through a combination of stereotactic and neuroimaging techniques. A subcutaneous external pacemaker which is internal programmable is implanted in the patients’ chest area. After the surgery, programming sessions are scheduled and adjustments on the pacemaker are made with respect to electrode configuration, voltage amplitude, pulse width, and frequency to send electrical currents to the brain for the best outcomes. Also all the benefits and adverse events are recorded to help the physicians make appropriate modifications.
2.3 Movement Disorders Center at UH Neurological Institute

Among the 25 centers nationwide, the Movement Disorders Center in the UH is recognized extensively for its treatment of Parkinson’s disease and DBS is one of the most innovative and effective technique. As the pioneer of DBS in North America, the team has
extensive experience to treat patients with PD in employing DBS. According to the clinical studies conducted by the Movement Disorders Center, at least 8 out of 10 Parkinson’s patients experience a significant improvement in functional ability with DBS [8].

For each patient, the Movement Disorders Center requires multiple clinical visits and various record of rating scales in order to ensure the best outcome for DBS patients. Unified Parkinson’s Disease Rating Scale (UPDRS) provides a longitudinal insight into patient’s disease progression. Tremor Rating Scale (TRS), Burke-Fahn-Marsden Dystonia Rating Scale (BFMDRS), Toronto Western Spasmodic Torticollis Rating Scale (TWSTRS) and Unified Dystonia Rating Scale (UDRS) are collected to assess the symptoms of PD. Mini Mental Status Exam (MMSE), Montreal Cognitive Assessment (MoCA) and Epworth Sleep Scale are recorded for cognitive and sleep assessments. In addition, there is Short Form (36) Health Survey to evaluate individual patients health status. All of the form need to be scientifically organized and managed in order to a successful DBS delivery.

Furthermore, there is an annual increase of 8000 to 10,000 new PD patients who will undergo the DBS treatment worldwide with approximately 60,000 who have undergone this treatment [22]. Each treatment can potentially generate over 1.5 million different parameters per lead for the electrodes per programming session and multiple programming sessions are carried out with various parameter settings after the surgery. Hence, as the number of patients interested and involved in DBS increases, larger volume of data would be generated, the managing of which is extremely crucial to the delivery quality of DBS procedure.
The previous data management system adopted by the Movement Disorders Center was for the use of generalized clinical data management and did not serve the best interest of the DBS procedure. The system lacks of bespoken functionality, customized interface and data management flexibility to be part of the standard operating protocol for DBS. It has becoming increasingly clear that a more comprehensive and customized environment is in great need of the Movement Disorders Center to effectively manage the entire DBS data lifecycle including data collection, retrieval, analysis and secured sharing within the institution.

2.4 Motivation

A successful DBS therapy involves accurate pre-surgery clinical data recording, careful selection of appropriate patients, precise implantation of the stimulation electrodes, and multiple post-surgery parameter settings and adjustments [10]. Through out the process, the management of the data quality, accessibility and legibility involves and affects almost every step in the procedure and therefore directly determines the quality of DBS care delivery.

There are clinical data capturing and management system existing on the market, however, no one is designed and specialized in DBS to scientifically document the intra-operative data including electrophysiological data and permanent electrode implantation location, or post-operative data like stimulation variables settings and adjustments associated with therapeutic benefit. Also, therefore no unified guidelines or off-the-shelf cookbook is
available for the design of DBS data management system. Furthermore, it is possible for multiple manufacturers and devices to emerge into the US market soon even if there is only one DBS manufacturer approved by the FDA currently. Assuming that the aforementioned devices do emerge, there will be no unified solution for all DBS manufactures or devices. This paper presents a web-based, non-manufacturer dependent DBS-CRADLE™.

DBS-CRADLE™ is designed as part of the standard operating protocol for DBS clinical care to manage both of superficial registry data and longitudinal clinical data. It mirrors the actual clinical work flow in the Movement Disorder Center by (1) tracking the referrals and outcomes together which facilitates new business; (2) coordinating patient care through the multidisciplinary evaluation process; (3) recording DBS programming reviews in relation to programming changes in order to achieve better adjusting settings. DBS-CRADLE™ has the following features:

1. Comply to DBS workflow by introducing core-specific data management.
2. Minimize artificial data entry mistake by utilizing built-in data validation mechanism.
3. Decrease information access time through built-in search mechanism and synopsis reports view.
4. Enhance system usability and consistency with user-friendly and unified interfaces.
5. Facilitate web page accessibility by providing the graphical navigation tool.
2.5 Organization of Thesis

This thesis presents the design of a web-Interface-driven electronic data capturing system for deep brain stimulation to support information capture, retrieval and analysis. DBS-CRADLE™ is a secure, efficient and comprehensive infrastructure that serves as a bridge between patients, physicians and other clinical fellows within the institution.

The rest of this thesis is arranged as follows. Chapter 2 discusses the software development and management; Chapter 3 explains the system architecture of DBS-CRADLE™; Chapter 4 discusses the notable features and outstanding techniques of the system; Chapter 5 explains more detailed about the system in terms of the user interface of each component; Chapter 6 proposes the possible future improvements and optimization; Chapter 7 generalizes a conclusion about DBS-CRADLE™.
Chapter 3

DBS-CRADLE™ Software Development

3.1 Objective

For decades, researchers and practitioners have analyzed how to successfully and efficiently manage software applications. Ever since 1985, the CHAOS Report of Standish Group [12] has been collecting case information on real-life IT environments and software development projects including web applications. Over the past 12 years, more than 50,000 completed IT projects have been scientifically studied and they are divided into three groups: success, challenged and impaired. The 3 top failure factors are discovered as following:

1. Incomplete requirement fulfillment. Projects that are impaired and ultimately canceled have an average of approximately 30% originally-specified features and functions flawed or incomplete according to the 1994 CHAO report [12].

2. Inadequate user involvement. User participation has a major effect on project resolution; in fact, in CHAOS report, user involvement is given 20% of success points, which is the number one contributor to project success [11].
3. Neglected agile process adoption. The projects lose their velocity and agility which are facilitated by adequate and iterative interactions between the development team and the end users without the adoption of agile process.

To avoid the previous pitfalls, this thesis proposes a software design objective for DBS data management system development: tuning into the customers’ needs through essential and sufficient developer-customer interactions and fulfilling the requirement in an innovative agile development strategy which offers a rapid and cost-effective development cycle.

Web-interface-driven design (WIDD), an unique version of agile strategy, is initiated and utilized in this thesis to accomplish the goal. WIDD enables a swift and flexible development process. Unlike traditional software development methods that are technology-centric, WIDD is user-centric and lets a variety of project stakeholders (including physicians and other future users) iteratively discover and assess the system by exercising the interfaces with the development team before any back end code is written. With instant and professional feedback from the clinical experts who have in-depth knowledge about DBS, the development team is able to incrementally develop and refine system interfaces till the complete customer satisfaction is achieved. Having the development team in tune with the clients both having the same understands of the goals has a major positive effect on a successful outcome [11]. Moreover, WIDD is both schedule and budget friendly. The joint use of graphical and verbal communication greatly expedites the development speed and reduces communication overhead between people from different disciplines by
using graphical communication. Especially when requirements change, traditional de-
velopment methods like “waterfall” and test-driven development which are both time and
effort costly [13] [14]. Agile process, on the contrary, has a much lower percentage of
time and cost overruns and has three times the success rate of the traditional development
method [15].

3.2 Agile Methodology

Throughout the life cycle of DBS-CRADLE™, the agile manifesto is used as a key
guidance. It is conducted in an iterative fashion as shown in Figure 3.1 following the four
core values [20]: (1) Individuals and interactions; (2) Working software; (3) Customer
collaboration; (4) Responding to change. The process starts with the development team
capturing the accurate high-level objective and requirements by working with the clinical
experts. Afterward, the development team make feasible strategies and plans for each
iteration. Then the team starts to prototype, design, develop and integrate each interfaces.
With each interface developed and tested, weekly meetings are held by the development
team to demonstrate the interfaces and exercise the functionality with the clinical fellows.
Instant client feedback and in-depth critiques are gathered in the meetings and iterative
improvements are made to the system afterwards until the clients are completely satisfied
with the presentation and usability of the interfaces. During each iteration, customers are
required to involve in half of the steps (the red peddles in Figure 3.1).
By adopting the agile process, the DBS-CRADLE™ Development team workforce is effectively allocated and tuned-in with customers. Through early meetings and rapid prototyping and development as well as timely adjustments and refinements, misinterpretations are minimized, priorities are clarified and difficulties are ironed out before most of the code is written. For the customers, frequent communicating with the technical team keeps them
well informed of the project’s progress rate. And they are able to involve and contribute to
the design and decision-making phases. This continual and sufficient interactions between
individuals facilitates the exchange of information and expedites the process of changing
and is the key factor to bring the development team, the end users and system all together.

Moreover, agile process ensures the project flexibility and adaptability during the lifes-
pan. Major evolution on requirements or scope occurs throughout the software develop-
ment life cycle. The agile process allows DBS-CRADLE™ not just to accommodate but
also to embrace the special situations. With organized personnel, documentation and cus-
tomer collaboration, DBS-CRADLE™ is able to discover alternate paths forward in ways
and react to unexpected changes not possible when other stringently-defined practices are
adopted.

### 3.3 Web-Interface-Driven Development

An effective clinical data management system should provide an intuitive and effective
interface, perform correctly-captured tasks, and output appropriate and timely feedback.
User interface, where almost all of the interactions between users and computers occur, is
undoubtedly the most important part of the system. A well-designed and fully-functional
interface provides a fertile opportunity that, if designed in accordance with the needs of the
user can lead to an optimal user experience. Taking precedence over accurate users needs and
client satisfaction, Web-Interface-Driven Development is an software development process
that relies on repetition of a short interface development cycle. It consists of 7 phases as shown in Figure 3.2.

WIDD starts with comprehensively collecting high-leveled requirements from the client for each module of the system. The user requirements usually consist of the system function, user experience expectation and database design. The development team will conduct thorough research and analysis to ensure customer needs are fully understood.

![Figure 3.2 WIDD Development Process](image)

The system interfaces is firstly designed and implemented without the corresponding database and back end code. Even if there is no actual data exchange between the application and server, the interfaces behave exactly like the system is fully developed. In the
weekly-based meetings, a variety of project stakeholders, who are the future users of the system, meet with the development team to exercise and assess the interfaces. Users representing different roles are able to see information and interact with the interfaces based on their privileges. By iteratively conducting users involvements, the development team is able to acquire immediate and professional feedback and suggestions and therefore achieve high quality deliverables by continuously refining the system. After multiple reversions of interfaces, the development team starts with the data model design and action controller implementation with a full understanding and an explicit goal of DBS-CRADLE™.

Till now, the development cycle is completed and the development team starts with another module using the same cycle. When each module is finished, it will be integrated into the system and tested both manually and automatically. When each component is running correctly and efficiently, the system is ready for operational deployment.

Compare to other development methods, Web-Interface-Driven Development allows the development team to:

1. Fully discover and clarify primary goal before most of code is written. This is the first step to avoid the pitfalls of soaring budget, delayed schedule, neglected quality or customer dissatisfaction. Understanding and achieving the objective is a key to successfully deliver a project.

2. Accurately convert verbal description to graphic representation. With development team and system users coming from diverse disciplines, the communication overhead and misinterpretation are expected. To overcome the potential communication
obstacle, instead of pure verbal presentation, DBS-CRADLE™ development team utilizes graphical interfaces as a bridge to demonstrates the system’s usability and functionality to the customer.

3. Frequently involve end users to obtain rapid and comprehensive feedback. WIDD allows the customers to be involved frequently and to be synchronized throughout the life span of the project. Also, the customer’s opinions and feedback are recorded and are valued by the development team as a guidance to optimize the system.

4. Iteratively improve design quality, making the next stage less costly to implement.

5. Reduce cost by catching and responding to changes in early stage. Changing a product early in the development process is a hundred times more expensive than modification in late stage when back-end has been fully programmed [16]. The modifications of the interface design and implementation requires a minimum amount of resource [16].
Chapter 4

The DBS-CRADLE™ DBS System

4.1 Ruby on Rails

DBS-CRADLE™ is implemented using Ruby on Rails’ Model-View-Controller (MVC) architectural pattern. Ruby on Rails (RoR) is an open-source, full-stack web application framework that is designed to support the implementation of web applications. RoR gathered multiple libraries useful for web development into a single cohesive software stack for web developers to use, which promotes the developers’ development productivity and ability. RoR emphasizes the following software engineering patterns and paradigms that makes it stand out:

1. Convention over configuration. RoR sets default ways to build a typical web application which increases development productivity and simplicity.

2. Don’t repeat yourself (DRY). DRY is a principle that aims at reducing repetition of information of all kinds, especially useful in multi-tier architectures.

3. Agile development. Less coding, more interaction and quicker response to changes make RoR work best with agile development.
4. Model-View-Controller (MVC). MVC decouples and interconnects model, view and controller in order to increase flexibility and reuse [18].

These characteristics not only increases the code readability, decreases maintenance difficulty, but also enables DBS-CRADLE™ to be seamlessly adapted to ever changing clinical environment. The implementation of RoR ensures that DBS-CRADLE™ has the feasibility and scalability to be adopted nationally even internationally to support patient information capture and share in DBS.

4.2 MVC Architectural Pattern

![MVC Architectural Pattern](image)

Figure 4.1 MVC Architectural Pattern
The MVC framework is a software architectural pattern for implementing a web-based system. It divides the system into three interconnected parts: model, view and controller to separate the functionality of code as shown in Figure 4.1. The controller accepts and routes incoming messages from web-browser to the appropriate actions. The model directly accesses and manages the database and provides logic, relationships, scopes, and instantiations of the underlying data. The view displays the result for the requesting web-browser using a template with embedded Ruby. The interactions between MVC are: the controller sends command to the models and views to update the state and presentation of the data; model notifies the associated controllers and views if changes are made; view makes requests to model in order to generate an updated presentation of information. The separation and cooperation of the three parts of MVC ensures the successful adoption of WIDD in DBS-CRADLE™.

4.3 System Environment

Table 4.1 illustrates the system environment required by DBS-CRADLE™. All the requirements listed below has been thoroughly tested. Users at UH are accessing DBS-CRADLE™ via Internet Explorer on windows machines. However, DBS-CRADLE™ is compatible with other web browsers like Google Chrome and FireFox on machines have either Mac OS or Microsoft Windows as the operation system.
### Table 4.1 DBS-CRADLE™ System Environment

<table>
<thead>
<tr>
<th>Name</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ruby</td>
<td>2.0.0</td>
</tr>
<tr>
<td>Rails</td>
<td>4.0.0</td>
</tr>
<tr>
<td>MySQL</td>
<td>5.6.15</td>
</tr>
<tr>
<td>Web Browser</td>
<td>Internet Explorer</td>
</tr>
<tr>
<td></td>
<td>Google Chrome</td>
</tr>
<tr>
<td></td>
<td>FireFox</td>
</tr>
<tr>
<td>OS</td>
<td>Mac OS</td>
</tr>
<tr>
<td></td>
<td>Microsoft Windows</td>
</tr>
</tbody>
</table>

**4.4 System Architecture**

DBS-CRADLE™ adopts 3-layer web application architecture from software perspective as shown in Figure 4.2. Users access the application through the presentation layer, which provides user interfaces, handles user interaction and validates user input. The business layer contains authentication and authorization mechanisms, handles information processing, makes logical decisions and performs calculations. It is also responsible for data exchange between two adjacent layers. The data layer stores and retrieves data from the database and manages data access. The presentation layer is running on end users’ computers; the business layer and the data layer runs on a separate server.
DBS-CRADLE™ supports various web browsers as end clients including Internet Explorer, Google Chrome and Firefox. DBS-CRADLE™ uses Apache server as the web server to handle HTTP requests sent from web browsers. MySQL is used as database management system which enables secure, scalable and high-performance Online Transaction Processing applications. When the end user sends a request through a web browser, a connection is established between the presentation layer and the business layer. Before the identification is authenticated, the users cannot gain access to database directly, which means patient information on the server is secure. Permitted users have different access level to patient data according to their roles. The 3-layer architecture ensures the data security and system scalability as well as execution speed.
4.5 Database Design

Figure 4.3 DBS-CRADLE™ Database Design
Patient data is navigated by 5 cores in DBS-CRADLE™. They are: administrative core, clinical core, surgery core, imaging core and adverse event core. There are a total of 30 data models for the 5 cores as shown in Figure 4.3. Administrative core contains patient demographic and registration information as well as care conference results. Clinical core includes PD diagnosis, clinical visits and physician evaluations. Surgery core holds surgery related data like DBS leads and stimulation sites. Imaging core stores the MRI or CT scans of patient’s brain. And adverse event core includes unexpected or undesired events resulting from DBS procedures.

### 4.6 Notable Features of DBS-CRADLE™

#### 4.6.1 Interactive Data Validation

DBS-CRADLE™ provides interactive data validation for user input to avoid, identify and correct errors and inconsistencies in patient data. A variety of implicit and explicit validation rules are used to check for correctness and meaningfulness of user input. The validation rules used in DBS-CRADLE™ includes:

1. Data presence check. This rule validates that the required fields are not left blank. For example, in the "Surgery" Core, fields marked with a red star like 'Date' and 6 of anterior commissure - posterior commissure (AC-PC) coordinates fields can not be empty. Otherwise, an warning message is displayed below the invalid field as shown in Figure 4.4.
2. Data range and limit constraint. In many data fields, valid input values have to be within a pre-determined range. For example, the rating scales in "Clinical Visit" Core have constraint on most of the numeric fields. A placeholder is displayed in the field to remind users of the valid range. Therefore, the field that violate the predefined validation rule are marked red as shown in Figure 4.5.

3. Data type conformance. Each data entry field is usually associated with a simple data type (integer, float, boolean, date, time) which can be used to automatically
apply data type conformance checks. In the example below the Systolic Blood Pressure field has been defined as an integer type which allow the electronic system to automatically check data entry for conformance against the type definition:

![Component Information](image1)

Figure 4.6 Data Type Validation Example

4. Data uniqueness guarantee. Specific fields like patient medical record number cannot be repeated in DBS-CRADLE™. If a user inputs a value that has been taken, an warning message would be displayed at the bottom of the field upon saving (Figure 4.7).

![Create Patient](image2)

Figure 4.7 Data Uniqueness Validation Example
5. Restricted value sets. Data entry fields may be associated with a set of permissible values. Drop down lists and radio buttons are used to restrict the selection. In Figure 4.8 the "Event" variable in a adverse event component is restricted to one of 37 values; no other data value can be entered by the user.

![Editing Adverse Event](image)

Figure 4.8 Restricted Value Set Example

The implementation of the data validation rules, (1) decreases the chance of potential human typing errors, (2) reduces the expensive and time-consuming post-entry artificial examination and correction; (3) increases data capturing accuracy which enables faster decision making.
4.6.2 Visual Navigation Aid

To enhance findability of the application and sections, DBS-CRADLE™ provides Bread-crumbs as the visual navigation aid. The navigation aid appears horizontally across the top of the interfaces, and below title bars, indicating the location of the user within the website’s hierarchy. The location information and links are in a backward linear manner; whereas, navigation methods, such as search fields or horizontal/vertical navigation bars, serve to retrieve information for the user in a forward-seeking approach. Figure 4.9 shows that the current user is viewing form 'SF36’ that was recorded on October 29th, 2014 in the Clinical Visit core of Test Patient’s medical information.

Figure 4.9 Visual Navigation Aid Example

The navigation aid does not only provides information to users as to where they are located within the application, but also offers shortcut links to all of the current page’s ancestors’ page in the hierarchical site structure without using the Back button, other navigation bars, or using keyword search.

4.6.3 Expandable and Collapsible Tables

A good interface communicates information in a concise, user-friendly way. DBS-CRADLE™ uses collapsible and expendable tables to ensure the readability and usability of an interface. In clinical visit core, forms like UPDRS (Unified Parkinson’s disease rating
scale) require a large volume of information in one interface. Instead of displaying the entire form, DBS-CRADLE\textsuperscript{TM} divides the form into several collapsible and expandable sub-forms as shown in Figure 4.10. Only one part of the form will be displayed at one time. Users can switch between the sub-forms by click the title of each sub-forms. This way allows user to access all of content of large tables in a uncluttered space without scroll up and down.

![Figure 4.10 Expandable and Collapsible Tables](image)

**4.6.4 Synopsis Report View**

By utilizing a synopsis report, all the generalized information can be viewed and trended in one screen. Interfaces like Clinical Visit consists of information about multiple forms. Instead overwhelming the users with every single detail, DBS-CRADLE\textsuperscript{TM} presents the
generalized information about each form created in a clinical visit as shown in Figure 4.11.

The user can click on the blue links to view detailed data of each form.

<table>
<thead>
<tr>
<th>Form</th>
<th>Setting (Med, Stim)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BFMDRS</td>
<td>(on, on)</td>
</tr>
<tr>
<td>TRS</td>
<td>(on, on)</td>
</tr>
<tr>
<td>MDS-UPDRS</td>
<td>(on, on)</td>
</tr>
<tr>
<td>Monopolar Review</td>
<td>L: Left STN L1, 2014-10-01, M: voltage</td>
</tr>
</tbody>
</table>

Figure 4.11  Partial Synopsis Report for Clinical Visit
Chapter 5

User Interface of DBS-CRADLE™

5.1 Authentication

DBS-CRADLE™ contains patient personal information and medical record, which under no circumstances should be revealed and maliciously accessed by unqualified personnel. To ensure the security of the data, the authentication procedure requires all the users to successfully log in to the system before getting access to the data as shown in Figure 5.1. A new user needs to provide personnel information like an email address, log in ID and contact info to successfully submit a registration for the use of DBS-CRADLE™ (Figure 5.2). Warning message will be displayed if any error or inappropriate format is made by the user.

Figure 5.1 Log in Page: only authorized users can access DBS-CRADLE™
User’s registration needs to be proved by any of the system administrators. Only authorized users can have access to DBS-CRADLE™.

Figure 5.2 Sign Up Page: every new user is required to signup in order to gain access
5.2 Index Page

Upon log in, the user is presented with a default page that consists of a list of created and saved patients list containing 6 columns, which are patient name, Medical Record Number (MRN), gender, last care conference date, last surgery date and primary diagnosis information (Figure 5.3). The patient list can be sorted in either ascendant and descendant order for any of the 6 columns. As default, the patient list is sorted in an ascendant fashion of the name column.

![Figure 5.3 Index Page: the list of existing patients](image)

There are 12 stages representing which phase each patient’s is in during the DBS procedure. They are: Interest in DBS, Neurologist Visit, DBS Neurologist Visit, Neurology Nurse, Neurosurgery Visit, MRI, Neuropsychologist Visit, Care Conference, Ready for
Surgery, Surgery, Programming, Neuropsychology Post-Operation. The 12 stages mirror the topological order of a DBS procedure, which allows the users to query the patient list by selecting their customized filter criteria according to the actual clinical flow.

Search function is implemented in DBS-CRADLE™ to help clinical fellows find the needed patient record efficiently. It can be triggered by searching the patient’s name and MRN. User can enter full or part of a patient’s name or medical record number in the text field, the patient list should be automatically narrowed down according to the input.

5.3 Create New Patient

Users with certain privileges can create new patients. By filling out the fields and finishing up by clicking the save button in Figure 5.4, a new patient is created and saved to the database.
Figure 5.4 Creation Page: the main page to create a new patient

<table>
<thead>
<tr>
<th><strong>First name</strong></th>
<th><strong>Last name</strong></th>
<th><strong>Middle initial</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Sex</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Male     Female</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Date of birth</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Form date</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>11/07/2014</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Ethnicity</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>--NR--</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Race</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>--NR--</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Education</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>--NR--</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Year of education</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Work status</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>--NR--</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Work reason</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>--NR--</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Marital status</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>--NR--</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Weight</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Height</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Diagnosis</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

- Movement Disorders
  - Hypokinetic
    - Parkinsonism
    - Cerebellar ataxia
  - Hyperkinetic
    - Tremor
    - Chorea
    - Tics
    - Dystonia

<table>
<thead>
<tr>
<th><strong>Handedness</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>--NR--</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Deceased</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Comments</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

**Save** or **cancel**
5.4 Core-specific Patient Data Management

Core-specific patient data management refers to the fashion that patient information is divided and managed by 5 different cores that accommodates the business logic and clinical flow of DBS procedure. The 5 cores are administrative core, clinical core, surgery core, imaging core and adverse event core. For easier patient data navigation, DBS-CRADLE™ provides 5 quick links to the cores and two additional links to the most frequency accessed patient information. The quick accessing links are profile, clinical visit, surgery, programming review, monopolar review, adverse even and imaging as shown in Figure 5.5.

![Figure 5.5 Patient Information Accessing Links](image)

5.4.1 Profile

Profile form includes 4 subforms: Demographic, Registration, Surgical Evaluation Workflow and Care Conference information (Figure 5.6). Demographic records patient demographic information, diagnosis, along with the record creation date. Registration subsection consists of the registration date as well as attending and referring physicians and fellows. Surgical evaluation workflow subsection contains pre-DBS and post-DBS evaluation of patient’s DBS procedure. Care conference stores patient’s qualification of DBS procedure and the acceptability of the MRI according to the patient’s clinical visit information and the evaluation from neurologist, neurosurgeon as well as neuropsychologist.
5.4.2 Clinical Visit

Clinical visit core records comprehensive data including a patient’s visit date, visit type, providers and pre-surgery eligibility assessments and post surgery result evaluations. There
are three different types of clinical visit and each of the clinical visit has distinctive components as shown in Table 5.1. Figure 5.7 shows three clinical visits made by Test Patient.

<table>
<thead>
<tr>
<th>Clinical Visit Type</th>
<th>Count</th>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neurologist Visit</td>
<td>8</td>
<td>Diagnosis, Genetics, Family History, Medications, Rating Scales, DBS, Adverse Events, Neurology DBS Evaluation</td>
</tr>
<tr>
<td>Neurosurgeon Visit</td>
<td>2</td>
<td>Neurosurgeon DBS Evaluation, Adverse Events</td>
</tr>
<tr>
<td>Neuropsychologist Visit</td>
<td>2</td>
<td>DBS Neuropsychology, Adverse Events</td>
</tr>
</tbody>
</table>

Table 5.1 Clinical Visit Types

Figure 5.7 Clinical Visit Page: overview of all clinical visits
5.4.3 Surgery

The surgery core stores 6 aspects of surgical data: surgery, record site, micro/macro stimulation site, OR contact threshold, IPG, battery and lead impedances (Figure 5.8).

Figure 5.8 Surgery Page: all information about DBS surgery
5.4.4 Programming Review

DBS programming involves various parameters including selection of optimal contacts, voltage, pulse width and frequency. There are more sophisticated options like interleaving and use of up to 2 programming groups for the best treatment outcomes. Programming review core is designed in a user-friendly and comprehensive way to capture and track the programming data along with notes on benefits and side effects. Figure 5.9 shows the index page of the programming review core. Most of the input fields provide a finite list of anticipated responses with a series of drop-down boxes presenting options. Other rows are configured to allow free text entry.

Figure 5.9 Programming Page: record programming parameters
5.4.5 Monopolar Review

Monopolar review is performed in order to determine the efficacy and side-effect thresholds for individual electrodes. A monopolar review records variables including the selection of cathode, amplitude, pulse width as well as benefit and side effects. Figure 5.10 shows the index page of the monopolar review core. Each monopolar review can be edited in an interface like Figure 5.11.

![Monopolar Review Page](image)

Figure 5.10 Monopolar Review Page
5.4.6 Adverse Event

The adverse event core collects information about undesired harmful effects resulting from DBS surgery. Figure 5.12 shows the index page of adverse event core.
5.4.7 Imaging

Imaging core stores the CT or MRI scan of the patient’s brain along with the date, scan type and scan parameters as shown in Figure 5.13. The imaging core assists the physicians to predetermine the implant location for the electrode leads, optimize the lead programming and investigate postoperative complications. Each scan can have multiple volumes which are 3-D pictures of patient brain that consists of a set of points.
Figure 5.13 Imaging Page: CT or MRI scan of the brain
Chapter 6

Result

DBS-CRADLE™ was developed in an iterative and client-centered fashion following WIDD which is an innovative agile software engineering methodology. The close, frequent and continuous interactions between clinical fellows and software developers enabled DBS-CRADLE™ to be implemented in a gradual manner interspersed with rapid modifications according to user feedback. As a result of the collaboration with the clinical experts at the Movement Disorders Center at the UH Neurological Institute in Cleveland, DBS-CRADLE™ provides a total of 38 distinct data capture interfaces for core-specific patient information management. Most of the interfaces have both edit mode and view mode. In the edit mode, there are 849 editable input fields for qualified uses to record information as shown in Table 6.1.
<table>
<thead>
<tr>
<th>Interface Name</th>
<th>Number of Input Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demographic</td>
<td>30</td>
</tr>
<tr>
<td>Registration</td>
<td>19</td>
</tr>
<tr>
<td>Care Conference</td>
<td>28</td>
</tr>
<tr>
<td>Surgical Evaluation Workflow</td>
<td>24</td>
</tr>
<tr>
<td>Genetics</td>
<td>10</td>
</tr>
<tr>
<td>Family History</td>
<td>13</td>
</tr>
<tr>
<td>Medications</td>
<td>10</td>
</tr>
<tr>
<td>UPDRS</td>
<td>76</td>
</tr>
<tr>
<td>MDS-UPDRS</td>
<td>60</td>
</tr>
<tr>
<td>TRS</td>
<td>56</td>
</tr>
<tr>
<td>BFMDRS</td>
<td>45</td>
</tr>
<tr>
<td>TWSTRS</td>
<td>36</td>
</tr>
<tr>
<td>UDRS</td>
<td>47</td>
</tr>
<tr>
<td>Epworth</td>
<td>17</td>
</tr>
<tr>
<td>MMSE</td>
<td>20</td>
</tr>
<tr>
<td>MOCA</td>
<td>12</td>
</tr>
<tr>
<td>SF36</td>
<td>153</td>
</tr>
<tr>
<td>Surgery</td>
<td>70</td>
</tr>
<tr>
<td>Record Site</td>
<td>10</td>
</tr>
<tr>
<td>Micro/ Macro Sti Site</td>
<td>20</td>
</tr>
<tr>
<td>OR Contact Threshold</td>
<td>20</td>
</tr>
<tr>
<td>IPG</td>
<td>7</td>
</tr>
<tr>
<td>Battery &amp; Lead Impedances</td>
<td>26</td>
</tr>
<tr>
<td>Programming Visit</td>
<td>30</td>
</tr>
<tr>
<td>Monopolar Review</td>
<td>10</td>
</tr>
<tr>
<td>Adverse Event</td>
<td>10</td>
</tr>
<tr>
<td>Imaging</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 6.1 Editable Input Fields for Each Editable Interface
Chapter 7

Conclusion

DBS has provided remarkable therapeutic benefits for patients with Parkinson’s disease. There is a growing interest and expansion in DBS surgery due to the high long-term patient satisfaction rate [21] and its wide application to other movement disorders. However, universal consensus on best practices for the surgical placement and medical management of DBS systems remains undefined.

DBS-CRADLE™ addresses this issue by serving as a flexible and scalable web-based application that actively captures clinical DBS data of patients who undergo the DBS procedure. It involves clinicians in the design and implementation of the system by using WIDD methodology; it provides various ways of data entry validation and assistance to avoid manual errors; it introduces patient stage concepts mirroring the actual clinical flow; it also helps users to efficiently navigate in the system by managing patient data with core concept.

DBS-CRADLE™ has been deployed at the UH Movement Disorder Center and preliminary user feedback shows that it has been serving its designed objectives. It can have a direct impact on clinical care decisions in the very near term on the following aspects
of clinical DBS practice: (1) the clinical database system could help standardize clinical visits and data collection, (2) identification of the target volume for therapeutic stimulation could improve the ability to plan the electrode trajectory pre-operatively, (3) visualization of the target relative to the patient anatomy could improve electrode placement decisions intra-operatively, (4) stimulation parameter selection could improve with our statistically driven decision support system.
APPENDIX  
Ruby on Rails Code

A.1 Sample Controller

```ruby
class MocasController < ApplicationController
  # GET /mocas
  # GET /mocas.json
  def index
    @mocas = Moca.all

    respond_to do |format|
      format.html # index.html.erb
      format.json { render json: @mocas }
    end
  end

  # GET /mocas/1
  # GET /mocas/1.json
  def show
    @moca = Moca.find(params[:id])

    respond_to do |format|
      format.html # show.html.erb
      format.json { render json: @moca }
    end
  end

  # GET /mocas/new
  # GET /mocas/new.json
  def new
    @moca = Moca.new
    respond_to do |format|
      format.html{ redirect_to '/', notice: 'The page you were trying to access does not exist' } # new.html.erb
      format.js { render 'edit' }
    end
  end
end
```
# GET /mocas/1/edit
def edit
  @moca = Moca.find(params[:id])
  respond_to do |format|
    format.html { redirect_to '/', notice: 'The page you were trying to access does not exist' }
    format.js { render 'edit' }
  end
end

# POST /mocas
# POST /mocas.json
def create
  @moca = Moca.new(params[:moca])
  respond_to do |format|
    if @moca.save
      format.js { render 'create', notice: 'Moca was successfully created.' }
      format.html { redirect_to '/', notice: 'The page you were trying to access does not exist' }
    else
      format.html { format.html { redirect_to '/', notice: 'The page you were trying to access does not exist' } }
      format.json { render action: "new" }
    end
  end
end

# PUT /mocas/1
# PUT /mocas/1.json
def update
  @moca = Moca.find(params[:id])
  respond_to do |format|
    if @moca.update_attributes(params[:moca])
      format.js { render 'update', notice: 'Moca was successfully Updated.' }
      format.html { redirect_to '/', notice: 'The page you were trying to access does not exist' }
    else
      format.html { redirect_to '/', notice: 'The page you were trying to access does not exist' }
      format.js { render 'edit', notice: 'Moca was not successfully Updated.' }
    end
  end
end
A.2 Sample Model

```ruby
# == Schema Information
#
# Table name: mocas
#
# id : integer not null, primary key
# mrn : integer
# visit_date : date
# date_entered : date
# education : string(255)
# s1 : integer
# s2 : integer
# s3 : integer
# s4 : integer
# s5 : integer
# s6 : integer
# s7 : integer
# s8 : integer
# s9 : integer
# s10 : integer
# s11 : integer
# s12 : integer
# s13 : integer
# s14 : integer
# s15 : integer
# s16 : integer
# s17 : integer
# s18 : integer
# s19 : integer
# s20 : integer
```
class Moca < ActiveRecord::Base
  attr_accessible :date_entered, :education, :last_edit_user_id, :mrn,
                  :patient_id, :s1, :s10, :s11, :s12, :s13, :s14, :s15, :s16, :s17,
                  :s18, :s19, :s2, :s20, :s21, :s22, :s23, :s24, :s25, :s26,
                  :s3, :s4, :s5, :s6, :s7, :s8, :s9, :total, :visit_date,
                  :word_1_category, :word_1_multi_choice, :word_2_category,
                  :word_2_multi_choice, :word_3_category, :word_3_multi_choice,
                  :word_4_category, :word_4_multi_choice, :word_5_category,
                  :word_5_multi_choice
  belongs_to :patient
  belongs_to :last_edit_user, class_name: 'User'
end

A.3 Sample View

<h3>Montreal Cognitive Assessment (MOCA) </h3>
<br />
<div style="width: 90%">
<table class="table table-simple">
  <tr>
    <td>MRN</td> <%= text_field_tag "mrn", "" %></tr>
    <td>Visit Date</td> <%= text_field_tag "visit_date", "" %></tr>
  </table>
</div>
<table>
<thead>
<tr>
<th>Visuospatial / Executive</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connect Trail</td>
<td>0−1</td>
</tr>
<tr>
<td>Copy Cube</td>
<td>0−1</td>
</tr>
<tr>
<td>Draw CLOCK</td>
<td>0−1</td>
</tr>
<tr>
<td>Naming</td>
<td>Score</td>
</tr>
<tr>
<td>Animal 1</td>
<td></td>
</tr>
<tr>
<td>Animal 1</td>
<td>Score</td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
</tr>
<tr>
<td>Animal 2</td>
<td>Score</td>
</tr>
<tr>
<td>Animal 3</td>
<td>Score</td>
</tr>
</tbody>
</table>

**Memory**
Read and Repeat List of Words. Two trials. Do a recall after 5 mins.
No Points

**Attention**
Read Numbers in Forward Order (1 digit/sec)
<table>
<thead>
<tr>
<th>Language</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repeat: I only know that John is the one to help today.</td>
<td></td>
</tr>
<tr>
<td>Repeat: The cat always hid under the couch when dogs were in the room.</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td></td>
</tr>
<tr>
<td>Fluency/ Name maximum number of words in one minute that begin with the letter F. (N ≥ 11 words)</td>
<td></td>
</tr>
<tr>
<td>Similarity between train - bicycle.</td>
<td></td>
</tr>
<tr>
<td>Similarity between watch - ruler.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S13</th>
<th>S14</th>
<th>S15</th>
<th>S16</th>
<th>S17</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Word 1</td>
<td>Category Cue</td>
<td>Multiple Choice Cue</td>
<td>Score</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>--------------</td>
<td>---------------------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Category Cue</td>
<td>Multiple Choice Cue</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Category Cue</td>
<td>Multiple Choice Cue</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Category Cue</td>
<td>Multiple Choice Cue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Category Cue</td>
<td>Multiple Choice Cue Cue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Word 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cue $S31$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cue $S32$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S33$, $S34$, $S35$, $S36$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Date, Month, Year, Day</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Orientation, Score</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
<td>
  <%= number_field_tag "S37", "", style: "width: 50px",
    placeholder => "0-1", min => "0", max => "1" %>
</td>
</tr>
<tr>
  <td>
    Place
  </td>
  <td>
    <%= number_field_tag "S38", "", style: "width: 50px",
    placeholder => "0-1", min => "0", max => "1" %>
  </td>
</tr>
<tr>
  <td>
    City
  </td>
  <td>
    <%= number_field_tag "S39", "", style: "width: 50px",
    placeholder => "0-1", min => "0", max => "1" %>
  </td>
</tr>
</div>
<tr>
  <td><b><h4>Total</h4></b></td>
  <td><%= text_field_tag "total", "", style: "width: 50px" %></td>
</tr>
</table>
LIST OF REFERENCES


[16] Snyder C. Paper prototyping: The fast and easy way to design and refine user interfaces. Boston; Amsterdam: Morgan Kaufmann; 2003


