Semi-automatic Database Design for Neuroscience Experiment Management Systems

Hao Li, BSc, James F. Brinkley, MD, PhD, John Gennari, PhD

Structural Informatics Group, Departments of Biological Structure and Medical Education and Biomedical Informatics, University of Washington, Seattle, WA 98195

Introduction

Neuroinformatics provides tools for neuroscience researchers to study brain function. In order to handle experiment paradigms that change frequently, we are developing a semi-automatic database design tool that will enable an experiment management system (EMS) to manage data with flexibility while retaining the efficiency of a relational database. In addition to accommodating heterogeneous data, the EMS will need to decrease application development and deployment time in order to mirror the rate at which experiment paradigms change in the laboratories.

Series of stimuli are shown to the patient in different modalities, such as picture, audio, or text.

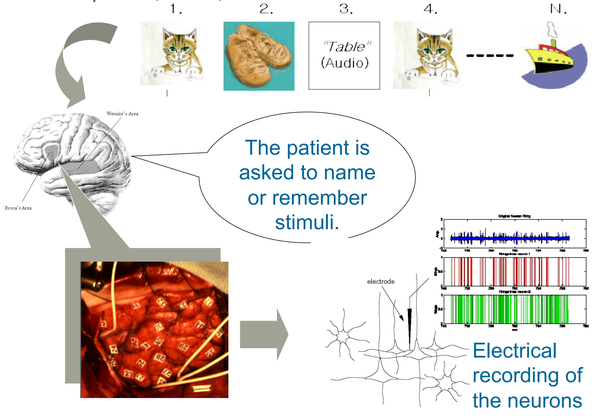
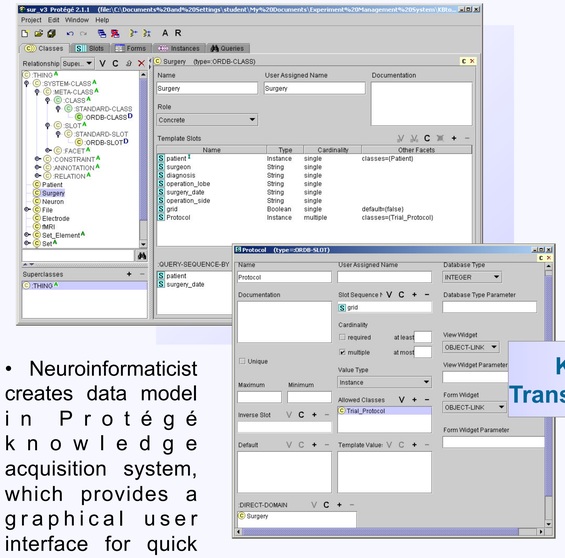


Figure 1. Basic data flow for a single unit recording experiment shows heterogeneous data sources.

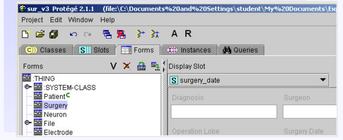
The Driving Application for the project is a single-cell recording (SCR) database developed for a neuroscience research group, which studies the correlation of human brain anatomy with language identification and memory ability. In these experiments, which are conducted during open brain surgeries, single neuron activity in cortical language areas is recorded and analyzed later for correlation with patient verbal responses (Figure 1).

System Design and Architecture

Figure 2.1. Data Model in Protégé



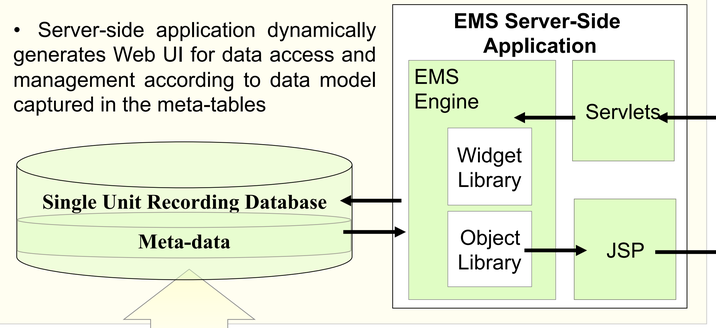
- Neuroinformaticist creates data model in Protégé knowledge acquisition system, which provides a graphical user interface for quick manipulation of complex data object relationships.
- Allows complex data relationships to be easily modeled and changed



- A new meta-class and a meta-slot provide additional information for generating relational database schema and the Web application descriptively. Examples are widgets used for different slots, whether the value of a slot is unique.

Figure 2.3. PostgreSQL database and Web server

- Server-side application dynamically generates Web UI for data access and management according to data model captured in the meta-tables



KB2DB Transformation

Object Relational Database Schema

Figure 2.2. Relational Database Schema Recaptures Protégé Data Model

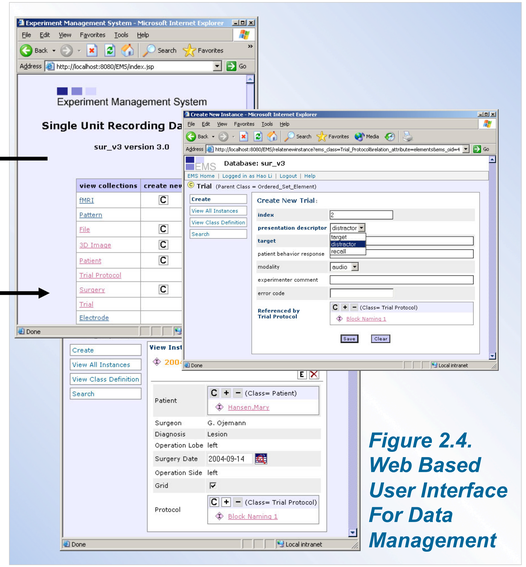


Figure 2.4. Web Based User Interface For Data Management

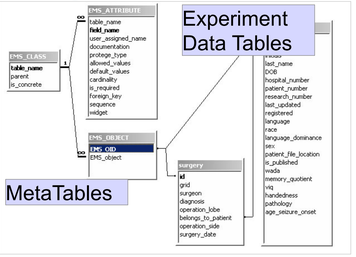
Sample SQL

```
CREATE TABLE Surgery (
    ID INTEGER UNIQUE NOT NULL PRIMARY KEY
REFERENCES _object ON DELETE CASCADE,
    patient INTEGER   DEFAULT NULL  ,
    surgeon VARCHAR(50)   DEFAULT NULL  ,
    diagnosis VARCHAR(50)   DEFAULT NULL  ,
    operation_lobe VARCHAR(50)   DEFAULT NULL ,
    surgery_date DATE   DEFAULT NULL  ,
    operation_side VARCHAR(50)   DEFAULT NULL  ,
    grid BOOLEAN   DEFAULT false
);

INSERT INTO _CLASS (_class, user_assigned_name,
direct_parent, is_concrete, browser_key,
sequence_by ) values ('Surgery', 'Surgery', ':THING',
true, 'Slot(surgery_date)', 'patient,surgery_date');

INSERT INTO _Attribute (_class, attribute_name,
user_assigned_name, documentation, protege_type,
database_type, allowed_values, default_values,
cardinality, is_required, foreign_key, widget,
view_widget, sequence ) values ('Surgery', 'patient',
'Patient', '', 'Instance', 'INTEGER', 'Patient', 'NULL', '1',
false, 'Patient', 'OBJECT_LINK', 'OBJECT_LINK', 0);
… … …
```

ER Diagram



- Semi-automatic transformation
- Database include relations for experimental data and meta-data that capture data model and Web UI specifications

Conclusion

The EMS application will facilitate flexible experiment paradigms through descriptive data modeling. Users can access and manage data in a relational database through this web-based user interface by simply creating a data model at the start of a project in Protégé. The data model is transformed to a relational database storing both meta data and data. A server-side application dynamically generates a web-based user interface based on the meta data model description of the data structure and look-and-feel of the user interface. We are in the first development cycle of the EMS application with each of the components described in the architecture in place. Our future goal is to enable the EMS application to adapt to frequent experiment changes.