Структурные особенности программного обеспечения для анализа результатов окулографических исследований

Г.А. Юрьев, А.С. Панфилова, П.А. Мармалюк

АННОТАЦИЯ

Рассмотрены особенности реализации и специфика структуры программного обеспечения предназначенного для компьютеризации и автоматизации расчётов, связанных с анализом данных психологических исследований в области глазодвигательной активности человека.

Приведены структура классов описание разрабатываемого программного обеспечения и структура базы данных, которая может использоваться для хранения экспериментальных и расчётных результатов. Описаны подходы к созданию графического интерфейса разрабатываемой программы.

КЛЮЧЕВЫЕ СЛОВА

Функциональное программирование, программное обеспечение, айтрекинг, язык программирования R, диаграмма классов, база данных, графический пользовательский интерфейс.

Structural features of software for oculography data analysis

G.A. Yuryev, A.S. Panfilova, P.A. Marmalyuk

Considered are implementation features and structural specificity of software, designed for computerization and automation of calculations related to psychological research in the field of eye movement analysis. The structure of developed software classes and its content, database structure that can be used to store experimental and computational results are under consideration.

KEYWORDS

Functional programming, software, eye tracking, R programming language, class diagram, database, graphical user interface.

Введение

Всё более широкое распространение для целей прикладного статистического анализа в мировом научном сообществе получает среда вычислений R, разработанная в 1993 году сотрудниками Оклендского университета [4]. Её популярность определяется рядом технических и исторических факторов. Среди технических факторов можно назвать удобную для математиков и других прикладных специалистов функциональную парадигму программирования. Она обеспечивает отсутствие побочных эффектов исполнения функций и субъективно более простую форму представления кода. Благодаря открытому программному коду, и возможности создания собственных обработчиков, широкое распространение получили пакеты, написанные сторонними разработчиками и размещённые в специализированном репозитории [4]. Такой подход позволил в очень короткие сроки увеличить возможности ядра системы.

Сейчас на российском рынке прикладного программного обеспечения для нужд статистики доминирующие позиции занимают закрытые среды, имеющие привычный для

Публикация подготовлена в рамках поддержанного РГНФ научного проекта №14-06-12012.

современного пользователя, графический интерфейс. В R используется интерфейс командной строки, что замедляет его распространение, в то же время плюсы в виде открытости программного обеспечения, кроссплатформенности и доступности прозрачных, готовых решений, постепенно увеличивают долю пользователей этого инструмента.

В данной статье авторы хотят представить проект находящегося в разработке пакета R, предназначенного для анализа данных окуломоторной активности испытуемых. Рассматриваются вопросы построения объектной структуры, используемой для хранения данных, а так же возможности связывания текущего проекта с базой данных (БД) и компонентами, обеспечивающими графический интерфейс разрабатываемому приложению.

Постановка задачи

В рамках своей профессиональной деятельности, экспериментальные психологи постоянно сталкиваются с задачами планирования постановки и проведения эксперимента. Финальным этапом экспериментального исследования, ради которого предпринимаются эти шаги, обычно является анализ собранных данных и подтверждение или опровержение гипотезы исследователя. С развитием технических средств и аппаратуры для сбора психофизиологических данных, психологи в своих исследованиях стали опираться, в частности, на данные, получаемые с применением специализированного оборудования.

Одним из перспективных направлений исследования сейчас являются эксперименты с применением оборудования для регистрации траекторий движения взора (айтрекеров). Данные, которые собирают с помощью этой аппаратуры, представляют собой (в подавляющем большинстве случаев) временные ряды, отражающие последовательность координат точек взора на плоскости визуального стимула. Дополнительно могут фиксироваться расширение зрачка и иные характеристики, вычисленные по первичным данным.

Традиционно, производители такого оборудования предоставляют и программное обеспечение, дающее определённые возможности по предварительной обработке и выгрузке данных произведённой записи. Это программное обеспечение часто является закрытым, а экспортированные данные требуют проведения дополнительного анализа, который затруднительно провести с помощью стандартных статистических пакетов. Авторы статьи ставят своей задачей разработку открытого программного обеспечения дающего широкие возможности по проведению анализа такого рода данных.

Структура классов

Как уже говорилось во введении, выбор среды для реализации проекта был обусловлен её гибкостью и широким распространением. В R (и во многих других прикладных системах) функциональная парадигма расширена рядом возможностей, характерных для императивных языков. Сюда можно отнести возможность прямого использования итераций, реализацию возможностей ООП и ряд нетривиальных приёмов, позволяющих обойти ограничения на функции с побочными действиями. Сохраняются и типичные плюсы функциональных языков, отсутствие (по умолчанию) побочных действий функций [2], возможность реализации функций высших порядков [1], наличие итераторов (встроенных функций высших порядков), механизмы лексических замыканий [3] и другие.

Для описания классов R использует нотацию языка S4 [21]. По умолчанию, структура такого класса предполагает наличие слотов (свойств) и, по необходимости, значений этих свойств, которые будут установлены при создании экземпляра. Методы же описываются за пределами тела класса («функционально-базированные методы» [9], вместо обычных для ООП «класс-базарованных» [9]) и представляют из себя специализированный вид функции, для которой программист может определить список

классов имеющих право её вызывать. Такой подход в данном случае служит одним из механизмов обеспечения ситуационного (ad hoc) полиморфизма, позволяя разным объектам иметь методы с одинаковым названием, но различной внутренней механикой работы.

Помимо этого, в соответствии с функциональной парадигмой разработки, методы не имеют доступа к свойствам (слотам) объекта из своего тела. Объект передаётся методу, как параметр, по значению. При необходимости внесения изменений в свойства этого объекта, после модификации, он возвращается в качестве результата исполнения функции и перезаписывается на место вызвавшего объекта. Второй вариант заключается в возвращении не всего объекта, а значения согласующегося по типу с одним из его слотов с последующей его перезаписью. Это нетипичное для ООП ограничение следует из принципа отсутствия побочного действия функций. Интерпретатор языка защищает глобальный контекст исполняемой программы от внешних вмешательств. С использованием специального расширения [22] это ограничение может быть преодолено при практической разработке.

Из сказанного следует, что приведённое далее описание структуры классов системы и её методов, является, в широком смысле, концептуальным представлением взаимодействия структур данных, а не отображением реального программного кода. То же касается и приведённой в репозитории диаграммы классов. По тем же причинам, в приведённых далее диаграммах методам часто передаётся сам объект, он же является возвращаемым значением. Описания методов и свойств сделаны в нотации C#, как одной из общепринятых. Все методы и свойства объявлены как публичные, что соответствует действительной ситуации с вызовом методов в R (отсутствуют модификаторы доступа).

Структура классов

Базовым классом, отвечающим за представление описания и результатов одного окулографического исследования является класс MetaExperiment, представленный на рисунке 1.

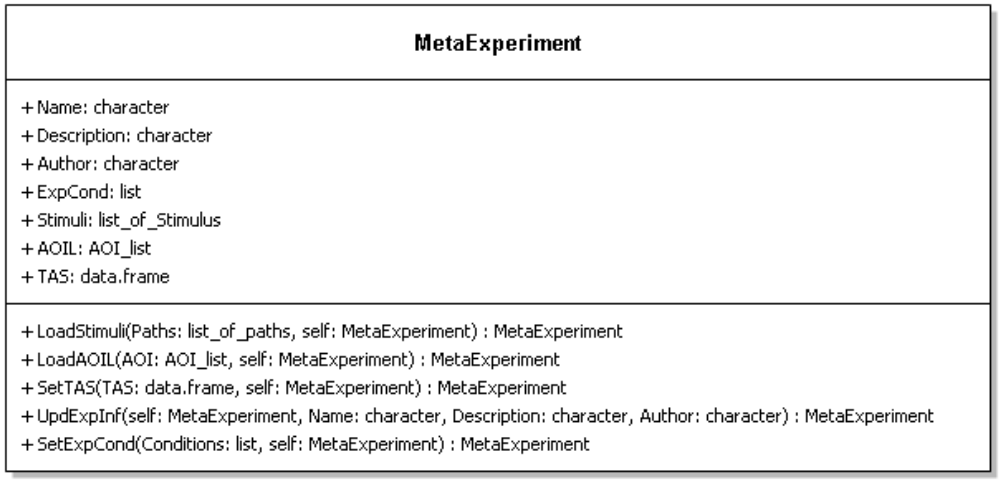

Рисунок 1. Структура базового класса MetaExperiment.

Содержательно этот класс представляет описание окулографического эксперимента, содержащие его название (Name), описание (Description), автора (Author), условия проведения эксперимента (ExpCond), список стимулов (Stimuli), список областей интереса (AOIL) и специализированную таблицу (TAS) устанавливающую связи между стимулами, зонами интереса и, возможно, иными переменными, зависящими от контекста эксперимента.

Подробнее стоит остановиться на понятии *условий эксперимента*. Как легко заметить, они (как и многие другие параметры) представлены списком, структура, содержание и способ адресации в котором могут существенно меняться. Такая типизация обеспечивает гибкость программного проекта, позволяя существенно модифицировать состав параметров функций, без значительного изменения программного кода. На данный момент, под условиями эксперимента понимаются такие параметры, как частота дискретизации записи, удалённость испытуемого от монитора, размеры и разрешение экрана, высота положения глаз испытуемого относительно нормали к центру экрана, особенности освещения и использованного при регистрации глазодвигательной активности оборудования. Эти данные будут необходимы при дальнейшей обработке. Можно заметить, что удобно представлять их при помощи различных типов (целочисленные переменные и переменные с плавающей запятой, а так же простые и составные типы). Объединение данных атрибутов в один ассоциированный список является удачным шагом на пути к созданию гибкого и прозрачного решения, при условии качественного оформления сопроводительной документации. При необходимости модифицировать или дополнить метод (функцию) использующий этот список в качестве одного из входов, список дополняется новым значением, а в теле метода делается необходимое дополнение.

Под *списком стимулов* понимается набор объектов, составляющих стимульный материал. В наиболее типичном случае это набор используемых в процессе эксперимента изображений, но на практике это могут быть кадры видеоряда (по сути, тоже изображения), задачи теста или часть экспериментальной ситуации.

Часто при проведении окулографических исследований, экспериментатора, помимо временного ряда координат взора испытуемого, интересует частота попадания взора в некоторую область на изображении (например, частота, с которой испытуемый в процессе эксперимента возвращался к исследованию области носа на продемонстрированном в качестве стимула портрете). Может представлять интерес и динамика движения взора в данной конкретной области или её частях, а сама область может задаваться достаточно сложным образом, от прямоугольника (в простейшем случае), до некоторой математической функции, определяющей границы *области интереса*. Возможны комбинации различных типов зон в рамках одного стимула. Эксперимент может подразумевать изменение этих зон в рамках одного стимула при возникновении какого-либо дополнительного условия. Таким условием может быть, например, восприятие фрустрирующего звукового стимула.

По сути, понятие зон интереса отражается в отношениях между следующими классами, представленными на рисунке 2.

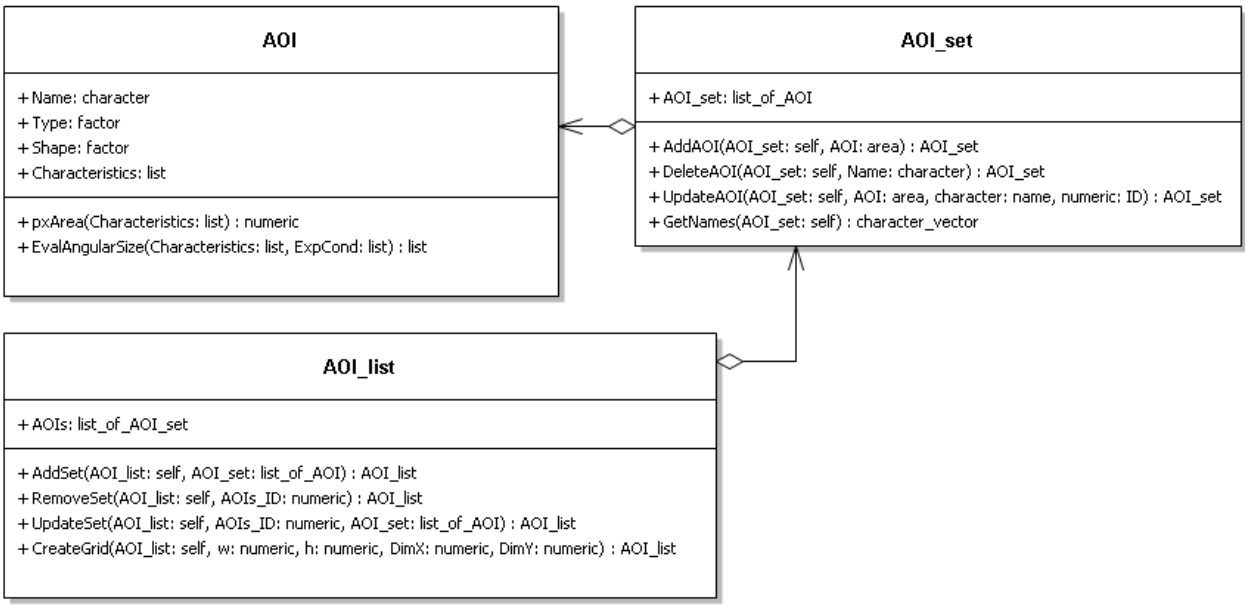

Рисунок 2. Структура отношений между классами определяющими зоны интереса и их содержание.

В данном случае AOI это класс, представляющий область интереса произвольного типа. Содержание его свойств меняется в зависимости от типа и формы необходимой зоны, которые определяются свойством Characteristics, являющегося списком. Например, в случае прямоугольной формы области, он может хранить координаты верхнего левого и нижнего правого углов, а в случае формы круга содержать его радиус и координаты центра.

Понятия *типа и формы* (Type и Shape) используются для определения логики обработки этого объекта. AOI_set это комбинация различных зон в рамках одного кадра, фактически список объектов класса AOI и методы работы с ним. Последний класс AOI_list это список всех комбинаций зон использованных в эксперименте, т.е. список объектов класса AOI_set.

Для установления *связей между зонами интереса, стимулами и экспериментальными ситуациями* (дополнительными, внешними условиями) предусмотрен слот TAS, являющийся таблицей, устанавливающей для каждой экспериментальной ситуации соответствие стимула, набора зон интереса и типа внешнего условия, определяющего изменение экспериментальной ситуации (если это необходимо).

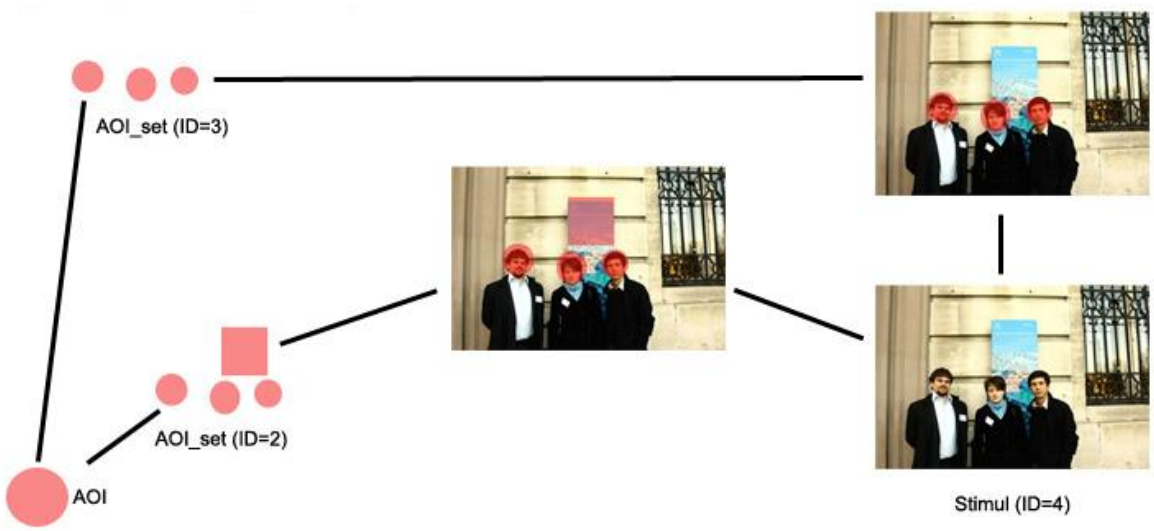

Рисунок 3. Иллюстрация отражающая смысл и контекст использования классов Stimul и AOI_set, а так же слота TAS.

На рисунке 3 представлены три различные экспериментальные ситуации, в рамках которых испытуемому помимо визуального стимула предъявлялся также звуковой:

1. Экспериментальная ситуация 1 – предъявление изображения трёх человек (стимул 4) на фоне звуков моря в наушниках;
2. Экспериментальная ситуация 2 – звук колокола, без смены изображений и списка областей интереса интереса (список областей 2);
3. Экспериментальная ситуация 3 - предъявление изображения трёх человек (стимул 4) на фоне звуков моря в наушниках, набор областей интереса меняется (список областей 3).

Первые две имеют идентичный набор зон интереса и отличаются только определённым внешним воздействием (звуком). В третьей ситуации, звук идентичен первой, но набор зон интереса меняется. Для всех трёх ситуаций используется один и тот же стимул. Фрагмент таблицы TAS (являющейся слотом класса MetaExperiment) для данного случая будет соответствовать таблице 1.

Таблица 1. Пример таблицы используемой для связи экспериментальных ситуаций, стимулов и зон интереса.

TrialID	AOI_setID	StimulID	TrialDescription
1	2	4	Предъявляется изо…
2	2	4	Звук колокола, без…
3	3	4	Предъявление изо…
...	...	...	...

От рассмотренного базового класса MetaExperiment, наследуется класс ExperimentData (рисунок 4), который расширяется дополнительными слотами, хранящими данные выборочного исследования. Предполагается, что это данные эксперимента, проведённого в соответствии с логикой, описанной в родительском классе.

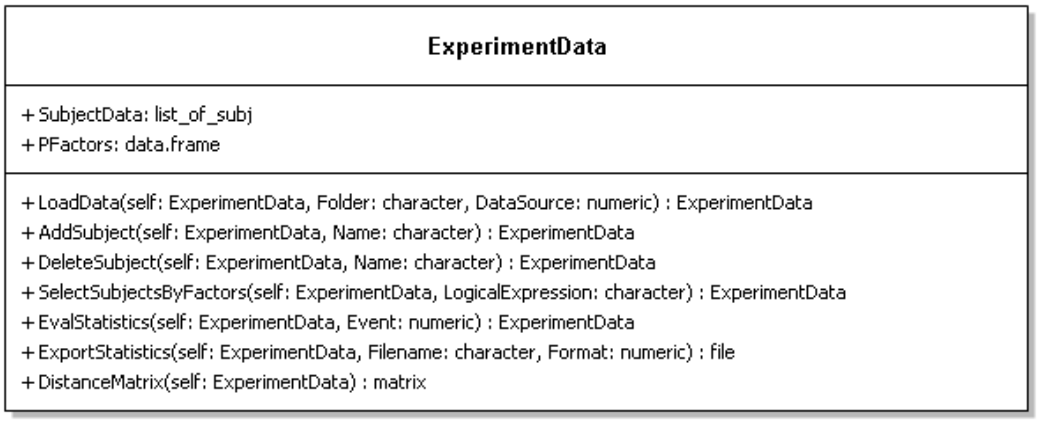

Рисунок 4. Структура класса MetaExperiment.

Дополнительные слоты – это список испытуемых (и собранных для них данных) и набор соответствующих этим субъектам факторов, которые могут представлять произвольную, выбранную исследователем, характеристику.

Данные участников эксперимента представляются списком объектов класса Subject, представленном на рисунке 5.

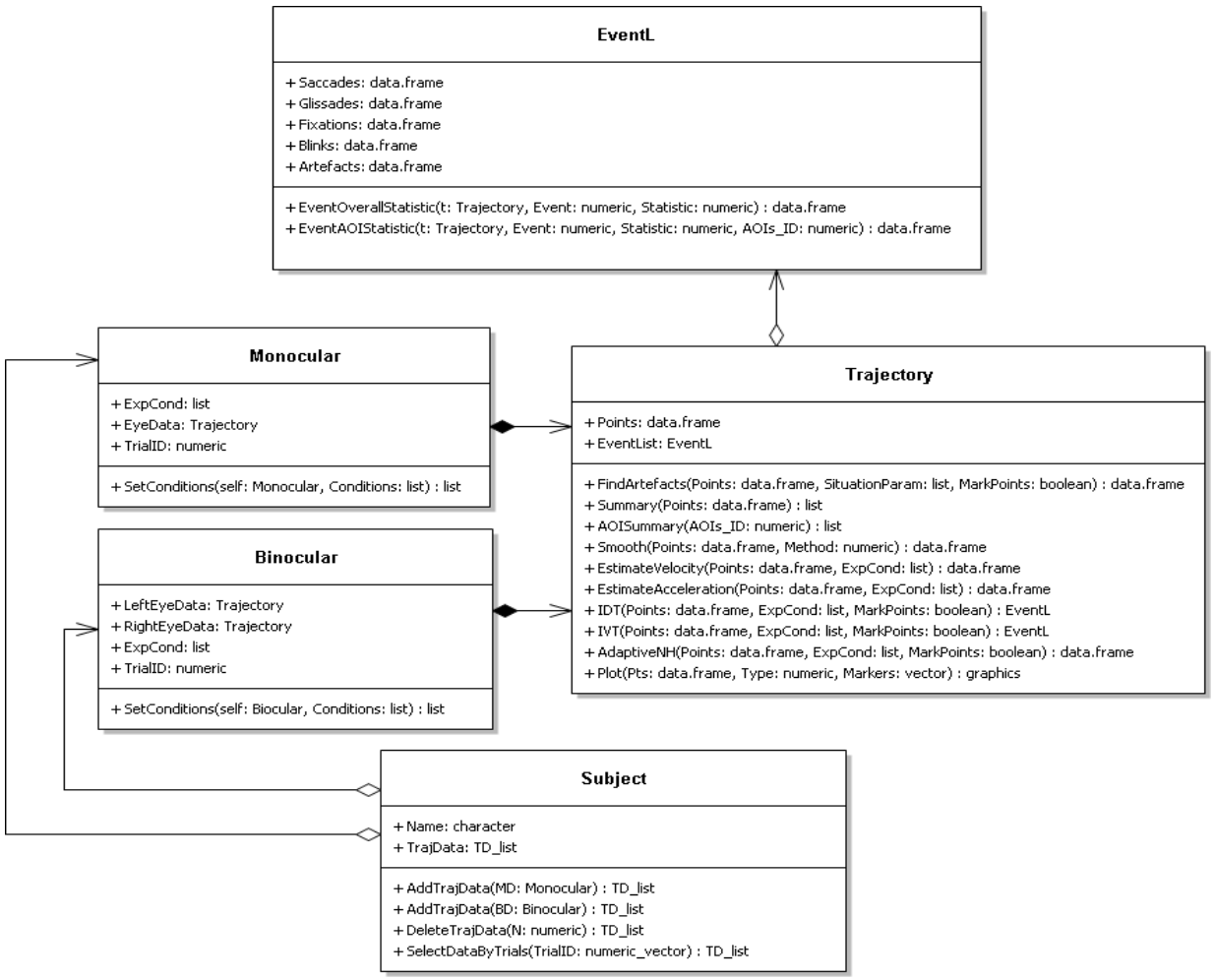

Рисунок 5. Структура отношений между классами представляющими испытуемых и результаты исследования.

Свойства этого класса определяют идентификационные данные участника исследования и список траекторий его взора, полученных в ходе исследования. Под траекторией в данном случае мы будем понимать не только сам временной ряд, но и дополнительные данные, связанные с ним.

Сам временной ряд хранится в форме объекта класса Trajectory, имеющего набор методов для выделения его характеристик. Данные исследования могут быть записаны с использованием различного оборудования, в частности, монокулярного (для фиксации траектории движения взора одного глаза) или бинокулярного (для фиксации данных по обоим глазам испытуемого). Эти ситуация отражена через объявление двух классов для этих случаев (monocular и binocular соответственно). Использование именно двух различных, независимых классов, вместо логичного в данном случае наследования одного от другого, обусловлено значительными различиями в логике обработки таких данных. При этом, оба класса содержат близкий набор свойств – это номер экспериментальной ситуации, одну или две траектории (в зависимости от использованного способа регистрации) и условия проведения эксперимента (ExpCond), которые могут быть либо взяты из общего описания условий, находящегося в рассмотренном ранее слоте класса MetaExperiment, либо переопределены вручную для конкретной траектории.

Траектория помимо временного ряда содержит свойство *список событий* (EventList), предназначенное для хранения информации о найденных по траектории окуломоторных событиях (артефакты записи, саккады, глиссады, фиксации, моргания).

Данное описание отражает взгляд авторов на декомпозицию задачи хранения и обработки данных окулографических исследований и последующее её решение в рамках среды программирования R.

Использование базы данных

Среда R позволяет использовать различные СУБД (PostgreSQL, MS Access, MS SQL Server, SQLite и MySQL) для работы с данными.

Обзор существующих решений для работы с базой данных

Рассмотрим основные библиотеки, обеспечивающие связь среды R с базой данных. Модуль PL/R позволяет внедрять функции на языке R в СУБД PostgreSQL, позволяющие задавать определенные правила для добавления, извлечения и удаления записей [6]. Использование MySQL возможно через пакет RMySQL [14], позволяющий создавать соединение с БД, составлять запросы на языке SQL, получать результаты выборки. RSQLite и sqldf [8] работают с одноименной базой данных. RSQLite отличает простота в использовании. При установке соединения с базой данных необходимо использовать команду

db <- *dbConnect(*SQLite(), *dbname*="Test.sqlite"*)*,

в которой требуется указать имя файла *.sqlite. Также данный пакет имеет удобный интерфейс для импорта данных из файлов формата CSV или XLS в базу данных. С помощью команды

dbWriteTable(*conn* = db, *name* = "Class", *value* = "class.csv", *row.names* = FALSE, *header* = TRUE)

есть возможность записать данные из файла (параметр *value*) в определенную таблицу базы данных (параметр *name*). Более универсальным является модуль RODBC [17], который позволяет использовать драйвер ODBC для подключения к различным базам данных с использованием следующей команды

channel <- *odbcConnect(*"test", *uid*="ripley", *pwd*="secret"*)*,

где первый параметр функции *odbcConnect()* обозначает кодовое название драйвера, используемого для подключения, параметры *uid* и *pwd* содержат информацию о логине и пароле, заданными для данной БД. Пакет поддерживает драйверы, представленные в таблице 2.

Таблица 2. Драйверы пакета RODBC.

Название параметра	**Драйвер**
test	MySQL ODBC 3.51 Driver
sqlite3	SQLite3 ODBC Driver
testpg	PostgreSQL ANSI
testacc	Microsoft Access Driver (*.mdb)
SQLServer	SQL Native Client
testpgw	PostgreSQL Unicode

RODBC позволяет проводить все необходимые операции с данными, имеет множество настроек и функций для удобной работы с результатом запроса.

Структура базы данных

Разработка структуры базы данных предполагает реализацию возможности хранения записей глазодвигательной активности для воспроизведения эксперимента и дальнейшего анализа. Основными сущностями разработанной структуры являются MetaExperiment и Subject, связанные между собой через таблицу отношений, позволяющей в дальнейшем сопоставить эксперименты и испытуемых, принимавших в них участие (рис. 6).

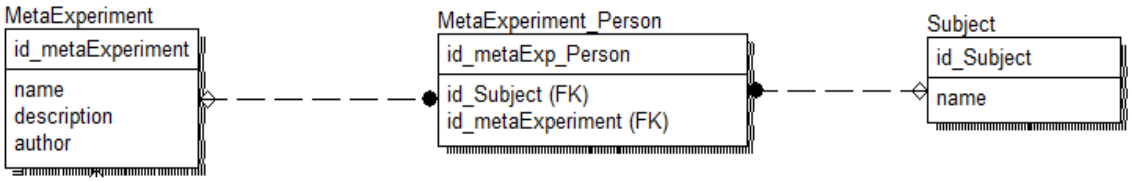

Рисунок 6. Структура отношений между таблицами испытуемых и настроек эксперимента.

Сущность MetaExperiment хранит описание окулографического эксперимента и содержит его название (name), описание (description), автора (author). Эксперимент связан с условиями его проведения через таблицу TRS, устанавливающей связь между стимулами (Stimulus) и списками зон интереса (AOI_list) (рис. 7).

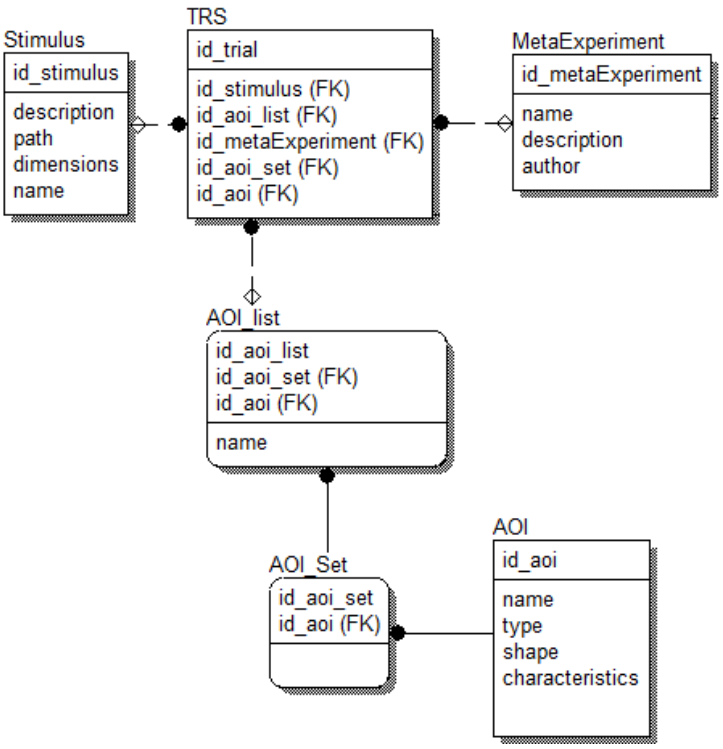

Рисунок 7. Структура отношений между таблицами стимулов, используемых в эксперименте, и зон интереса.

Подробное рассмотрение понятия областей интереса представлено в п. «Структура классов». Разработанная схема предполагает реализацию возможности хранения данных эксперимента с описанием используемых стимулов, соответствующих им зон интереса, представляемых с помощью различных объектов, для возможности дальнейшего воспроизведения эксперимента, а также соотнесения параметров глазодвигательной активности с конкретной экспериментальной ситуацией.

Для обеспечения возможности извлечения данных об условиях проведения эксперимента для конкретного испытуемого используются сущности MetaExp_Person_ExpCondition и PFactors (рис. 8).

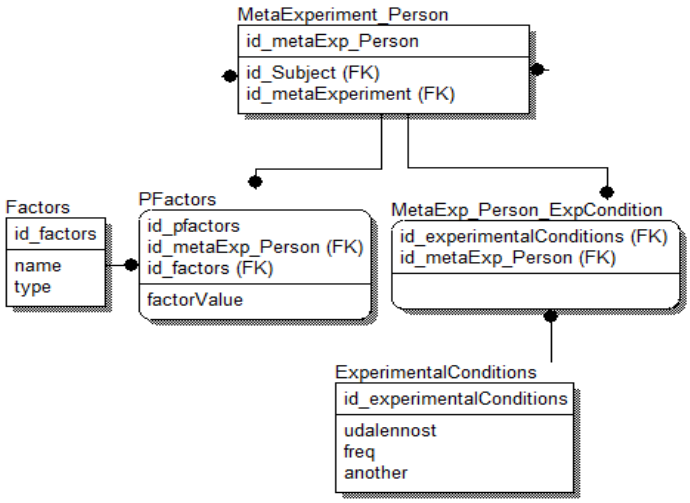

Рисунок 8. Структура отношений между таблицами, которые характеризуют испытуемых в рамках эксперимента и условия проведения исследования.

Зависимая сущность PFactors содержит номер фактора (id_factors) из таблицы Factors и его значение (factorValue). В качестве факторов может выступать результат отнесения испытуемого к какому-либо классу рамках проведенного эксперимента, а также другие характеристики испытуемого, необходимые для данного эксперимента. Зависимая сущность MetaExp_Preson_ExpCondition обеспечивает соотнесение параметров проведения заданного эксперимента (ExperimentalConditions): удаленность испытуемого от экрана (udalennost), частота дискретизации айтрекера (freq), другие простые атрибуты (another), - с определенным испытуемым.

Проведение окулографического эксперимента возможно, как упоминалось ранее, с использованием двух вариантов записи глазодвигательной активности: монокулярной и бинокулярной, - для которых обозначены таблицы Monocular и Binocular, сопоставляемые определенному испытуемому через сущность TrajData.

Хранение в базе данных записей о глазодвигательной активности нецелесообразно, так как эта информация достаточно велика по объему и зависит от числа фиксируемых параметров, длительности записи и частоты дискретизации. Например, при частоте дискретизации в 1000Гц, 100 записей в монокулярном режиме, каждая из которых длится 3 минуты, будут содержать более 18 миллионов значений. Авторами предлагается хранить в сущности Trajectory ссылку на файл с записями эксперимента, а также информацию о формате получаемых данных для возможности их дальнейшего конвертирования в формат разрабатываемой системы. Таблица Trajectory связана с сущностью TRS по типу один-ко-многим для возможности соотнесения траектории с определенной экспериментальной ситуацией. Связь с сущностями Left и Right необходима для дальнейшего анализа данных, требующего информацию о записях правого и левого глаза в отдельности. Предлагаемая структура хранения информации о траекториях, позволяющая соотносить ее с испытуемыми и различными видами записей глазодвигательной активности, представлена на рис. 9.

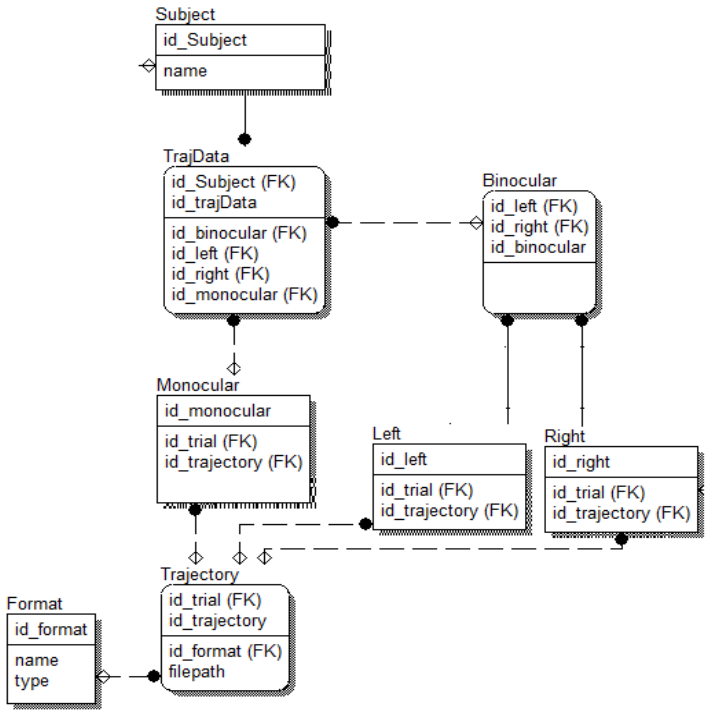

Рисунок 9. Структура отношений между таблицами, описывающими информацию о траектории и типах записей.

Информация о таких событиях, как моргание (Blink), глиссада (glissade), саккада (Saccade), фиксация (Fixation) и артефакты (Artefact), может быть выделена в рамках анализа монокулярных и бинокулярных траекторий. Данные события могут быть соотнесены с траекториями, записанными в монокулярном режиме или в отдельности для правого и левого глаза, через сущность Event (рис. 10).

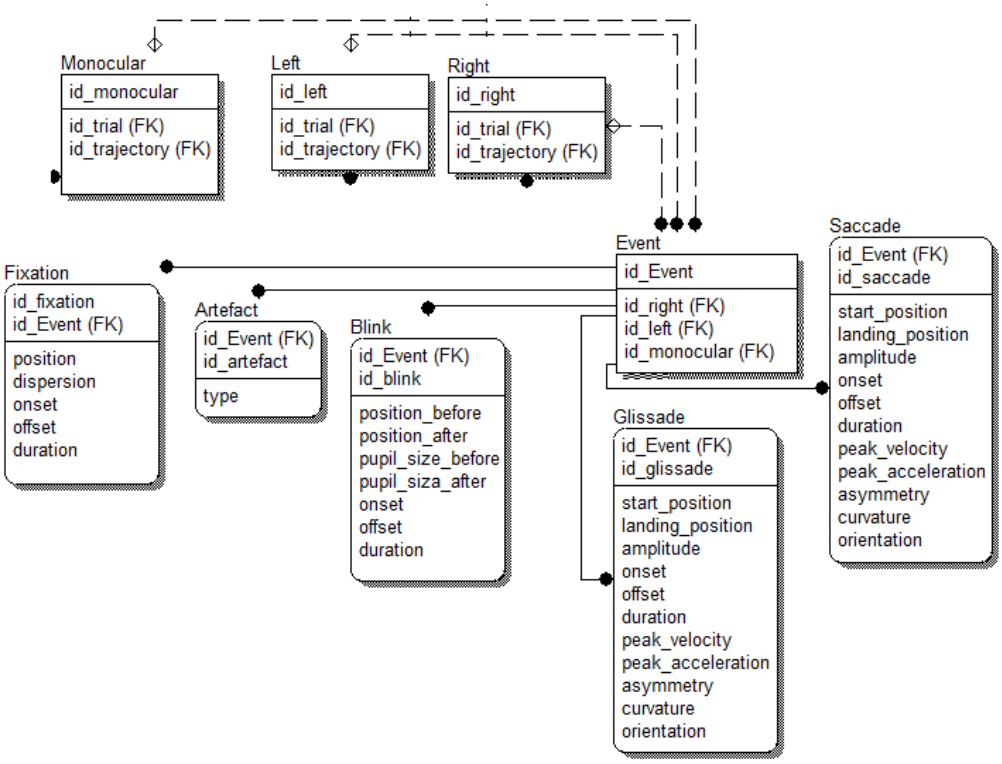

Рисунок 10. Структура отношений между таблицами, описывающими режим записи траекторий, и выделенными событиями.

Таким образом, разработанная авторами структура базы данных позволяет извлекать информацию о траекториях, относящихся к определенной экспериментальной ситуации, сопровождающейся демонстрацией стимулов и относящихся к ним областей интереса. Данную информацию можно сопоставить с испытуемыми, принимавшими участие в исследовании, их характеристиками и параметрами исследования, обозначенными для каждого тестируемого.

Графический интерфейс

Разработка системы анализа результатов окулографических исследований предполагает реализацию большого числа математических методов, результаты применения которых в большинстве случаев желательно представлять графически. Также необходимо предоставить пользователю возможность работы с данными траекторий, экспериментов и испытуемых в интерактивном режиме. Представляет интерес реализация возможности задания зон интереса для стимулов в режиме drag-and-drop (перетаскивание). Данные требования предполагают реализацию графического интерфейса пользователя и использование языка R изначально предполагало работу в режиме командной строки, но на данный момент доступен ряд графических интерфейсов пользователя, например, пакеты: RCommander, RKWard, RStudio.

RCommander [10] реализован в пакете Rcmdr [11] и в качестве отдельного приложения. При загрузке данного модуля, пользователь работает в модальном окне среды R и получает доступ к меню, которое позволяет импортировать данные разных форматов, проводить статистический анализ, строить различные графики, модели и др. (рис. 11).

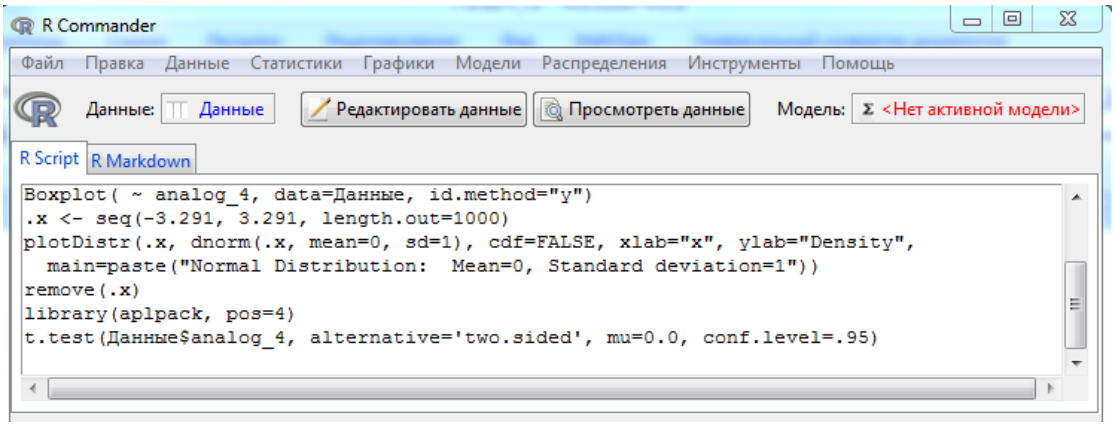

Рисунок 11. Интерфейс модуля R Commander.

Данный пакет предлагает возможность добавления дополнительных пунктов меню, создание диалоговых и модальных окон. Однако, данный модуль продемонстрировал нестабильность в работе.

RKWard [18] представляет собой отдельный программный продукт, который использует для работы библиотеки от среды KDE. В RKWard присутствует браузер текущего окружения, редактор данных. Он умеет перехватывать создание графических окон и добавлять к ним функции сохранения содержимого графического окна в файлы популярных форматов (PDF, EPS, JPEG, PNG). Интерфейс RKWard достаточно гибкий: пользователь может его расширять за счёт написания плагинов (собственно, все встроенные средства анализа, доступные сразу же после запуска из меню это тоже плагины, но только написанные авторами RKWard) [5]. Представленная система обладает рядом недостатков, среди которых можно выделить следующие:

- необходимость установки KDE;
- необходимость установки среды RKWard;
- сложность в расширении функционала.

RStudio [19] доступна в двух версиях: RStudio Desktop, в которой программа выполняется на локальной машине как обычное приложение; и RStudio Server, в которой предоставляется доступ через браузер к RStudio установленной на удаленном Linux-сервере. RStudio написана на языке программирования C++ и использует фреймворк Qt для графического интерфейса пользователя. Rstudio представляет большой набор дополнительных функций для работы в командной строке: управление файлами, завершение кода, автоматическое создание функций, инструменты навигации по коду, контроль версий. Инструмент Shiny [20] позволяет создавать веб-приложения с использованием языка R и HTMLL, CSS, JavaScript, что дает большие возможности для реализации интерфейса пользователя и вывода графики в интерактивном режиме. Недостатком пакета Shiny является необходимость использования веб-сервера с установленным программным продуктом RStudio Server, что потребует загрузки файла с результатами эксперимента, который может достигать 1Гб, в интернет.

В связи с тем, что существующие решения зачастую требуют установки дополнительного ПО, сложны в доработке и расширении, необходимо разработать графический интерфейс пользователя, позволяющий выполнять необходимую работу с данными, способный работать в приложении R, предоставляемым разработчиками.

В результате анализа определены ряд пакетов, которые наиболее часто используются для разработки GUI. Пакеты RGTk2 [15, 16] и tcltk2 [12] позволяют создавать модальные окна с пунктами меню, диалоговые окна, окна загрузки/сохранения файлов, использовать различные элементы управления, создавать табличное представление данных. Пакет gWidgets2 [13, 23] расширяет функционал вышеупомянутых модулей и предлагает удобный интерфейс для разработки пользовательского интерфейса.

Заключение

В статье рассмотрена специфика разработки программного обеспечение в рамках языка R, пути создания графического интерфейса программы. Приведены диаграмма классов разрабатываемого программного обеспечения и возможная структура базы данных для хранения описания и результатов экспериментов проведённых с использованием окулографического оборудования. В качестве следующего шага, авторы видят реализацию основного функционала программного обеспечения в соответствии с описанной концепцией, его развитие и расширение.

Список использованных источников

1. Вольфенгаген В.Э. Конструкции языков программирования. Приемы описания.– М.: АО “Центр ЮрИнфоР”, 2001. – 276 с.
2. Жиль Довек, Жан-Жак Леви. Введение в теорию языков программирования – ДМК Пресс, 2013. -134 с.

3. Люгер, Джордж, Ф. Искусственный интеллект: стратегии и методы решения сложных проблем, 4-е издание. : Пер. с англ. - М.: Издательский дом 'Вильяме', 2003. - 864 с.
4. Статистический анализ данных в системе R. Учебное пособие / А.Г. Буховец, П.В. Москалев, В.П. Богатова, Т.Я. Бирючинская; Под ред. проф. Буховца А.Г. - Воронеж: ВГАУ, 2010. - 124 с.
5. Шипунов А.Б., Балдин Е.М., Волкова П.А., Коробейников А.И., Назаров С.А., Петров С.В., Суфиянов В.Г. Наглядная статистика. Используем в R! – М.: ДМК Пресс, 2014. – 298с.
6. Conway J.E. PL/R User's Guide - R Procedural Language. – URL: http://www.joeconway.com/plr/doc/plr-US.pdf (дата обращения: 10.07.2014).
7. CRAN - The Comprehensive R Archive Network [Электронный ресурс]. – URL: http://cran.r-project.org/web/packages/em2/index.html (дата обращения 27.06.2014).
8. David A. James, Falcon S. RSQLite. – URL: http://cran.r-project.org/web/packages/RSQLite/RSQLite.pdf (дата обращения: 10.07.2014).
9. ECOOP '92. European Conference on Object-Oriented Programming: Utrecht, The Netherlands, June 29 - July 3, 1992. Proceedings (Lecture Notes in Computer Science) – Springer, 1992. – 429 pp.
10. Fox J. The R Commander: A Basic-Statistics Graphical User Interface to R/ Journal of Statistical Software, 2005. – 14(9). – URL: http://www.jstatsoft.org/v14/i09/paper (дата обращения: 10.07.2014).
11. Fox J., Bouchet-Valat M. Rcmdr. – URL: http://cran.r-project.org/web/packages/Rcmdr/Rcmdr.pdf (дата обращения: 10.07.2014).
12. Grosjean P. tcltk2. – URL: http://cran.r-project.org/web/packages/tcltk2/tcltk2.pdf (дата обращения: 10.07.2014).
13. Huang B., Cook D., Wickham H. tourrGui: A gWidgets GUI for the Tour to Explore High-Dimensional Data Using Low-DimensionalProjections/ Journal of Statistical Software, 2012. – 49(6) . – URL: http://www.jstatsoft.org/v49/i06/paper (дата обращения: 10.07.2014).
14. James D. A., DebRoy S. RMySQL. – URL: http://cran.r-project.org/web/packages/RMySQL/RMySQL.pdf (дата обращения: 10.07.2014).
15. Lawrence M., Lang D.T. RGtk2: A Graphical User Interface Toolkit for R/ Journal of Statistical Software, 2010. – 37(8). – URL: http://www.jstatsoft.org/v37/i08/paper (дата обращения: 10.07.2014).
16. Lawrence M., Lang D.T. RGtk2. – URL: http://cran.r-project.org/web/packages/RGtk2/RGtk2.pdf (дата обращения: 10.07.2014).
17. Ripley B., Lapsley M. RODBC. – URL: http://cran.r-project.org/web/packages/RODBC/RODBC.pdf (дата обращения: 10.07.2014).
18. Rodiger S., Friedrichsmeier T., Kapat P., Michalke M. RKWard: A Comprehensive Graphical User Interface and Integrated Development Environment for Statistical Analysis with R/ Journal of Statistical Software, 2012. – 49(9) . – URL: http://www.jstatsoft.org/v49/i09/paper (дата обращения: 10.07.2014).
19. Rstudio [Электронный ресурс]. – URL: http://www.rstudio.com (дата обращения: 10.07.2014).
20. RStudio, Inc. shiny. – URL: http://cran.r-project.org/web/packages/shiny/shiny.pdf (дата обращения: 10.07.2014).
21. Seamless R and C++ Integration with Rcpp. Series: Use R!, Vol. 64. Dirk Eddelbuettel, Romain Francois, 2013 - Springer, 220 p.
22. Seminar for statistics (SfS) [электронный ресурс]. – URL: http://stat.ethz.ch/R-manual/R-devel/library/methods/html/refClass.html (дата обращения 27.06.2014).
23. Verzani J. gWidgets2. – URL: http://cran.r-project.org/web/packages/gWidgets2/gWidgets2.pdf (дата обращения: 10.07.2014).