ГЛАВА 2

Разработка алгоритма работы автоматизированной системы

На начальном этапе создания автоматизированной системы следует разработать алгоритм ее работы. Здесь стоит отметить, что под алгоритмом работы автоматизированной системы будет подразумеваться алгоритм работы программного обеспечения, выполняющего соответствующие функции. Также стоит уточнить, что в данном контексте в качестве алгоритма будет рассматриваться последовательность инструкций, выполняемых в рамках одного стабилографического измерения – разумеется, стабилографическое исследование может включать в себя более одного измерения.

В соответствии с алгоритмом проведения типичного стабилографического измерения, приведенного в главе 1, а также принимая во внимание выявленные функциональные и нефункциональные требования к проектируемой автоматизируемой системе, приведенные в той же главе, был разработан следующий алгоритм работы системы.

- Предварительные действия: снятие показаний с аппаратно-программного комплекса «Многофункциональное кресло» при помощи персонального компьютера с использованием программного обеспечения StabMed версии 2.09 и выше, результат – стабилограмма (сигнал).

- Запуск проектируемой автоматизированной системы обработки результатов стабилографических измерений.

- Формирование и отображение графического пользовательского интерфейса автоматизированной системы.

- Получение системой данных (сигналов) из StabMed.

- Математическая обработка полученных сигналов.

- Анализ обработанных данных на предмет наличия аномалий в сигнале, которые предположительно могут являться маркерами патологий нервной системы и/или опорно-двигательного аппарат испытуемого.

- Визуализация обработанных стабилограмм и результатов их анализа.

- Классификация данных как носящих либо не носящих патологический характер экспертом (исследователем-физиологом).

- Сохранение классифицированной, обработанной предусмотренными системой математическими методами стабилограммы.

- Останов автоматизированной системы.

Схема данного алгоритма приведена на рисунке .

Проектирование структуры автоматизированной системы

На основании приведенного выше алгоритма можно произвести декомпозицию будущей автоматизированной системы на автономные подсистемы, выполняющие определенную функцию в рамках алгоритма работы. Таким образом проектирование и реализация значительно упрощаются. Данный подход также упрощает и повышает эффективность тестирования системы, поскольку появляется возможность начать тестирование на более раннем этапе проектирования и реализации. Большей эффективности также будет способствовать то, что на ограниченный функционал подсистемы единовременно потребуется меньше тестов, что облегчит обнаружение дефектов.

На основании алгоритма работы, представленного на рисунке , можно выделить следующие составные части автоматизированной системы:

- интерфейс Component Object Model;

- модуль математической обработки сигналов;

- модуль детектирования аномалий;

- графический пользовательский интерфейс;

- модуль визуализации стабилограмм;

- хранилище измерений.

Структурная схема автоматизированной системы обработки стабилографических измерений представлена на рисунке .

Разработка интерфейса по стандарту Component Object Model

Component Object Model (COM, компонентная объектная модель) - это стандарт интерфейса для программных компонентов, созданный компанией Microsoft. Он используется для создания объектов межпроцессного взаимодействия с использованием большого количества различных языков программирования. COM является основой для нескольких других технологий и платформ Microsoft, включая OLE, OLE Automation, Browser Helper Object, ActiveX, COM +, DCOM, DirectX и некоторых других. Суть COM - это независимый от языка способ реализации объектов, который можно использовать в средах, отличных от той, в которой они были созданы, даже за пределами машин. Качественно написанные компоненты COM позволяют повторно использовать объекты при отсутствии сведений об их внутренней реализации, вынуждая разработчиков компонентов предоставлять четко определенные интерфейсы, которые отделены от реализации.

Концепция объектов в COM похожа на концепцию объектно-ориентированного программирования, являясь, по сути, ее более универсальным вариантом. При реализации Component Object Model объекты располагаются в отдельных исполняемых модулях. Такими модулями могут быть либо файлы Dynamic-link library (DLL), либо непосредственно EXE-приложение.

При этом Component Object Model не является ни языком программирования, ни библиотекой, ни компилятором – это набор правил проектирования программ, которые будут вести себя определенным образом и иметь специфические возможности.

Основу модели COM составляют три принципа:

1. Независимость от контекста и уникальность компонентов: компоненты с одинаковым способом обращения имеют одинаковый смысл и назначение.
2. Инкапсуляция: внутреннее устройство COM-объектов скрыто для обеспечения независимости от языков программирования.
3. Доступность и портативность: необходимый компонент можно быстро найти динамически без привязки к файловой системе.

Таким образом преимуществом Component Object Model является наличие у каждого разработчика возможности кроссплатформенного обмена информацией между разрабатываемыми модулями, без привязки к средствам реализации и языкам программирования.

В модели COM взаимодействие между компонентами выстраивается по схеме «Клиент-сервер»: существует исполняемый модуль, в функционал которого включено исполнение некой инструкции, и приложение-инициатор, которому необходимо исполнить данную инструкцию, но которое «не знает», как это сделать, «зная» при этом о существовании исполняемого модуля и обращаясь к нему при возникновении такой необходимости; первый компонент будет являться «сервером», а второй – «клиентом». Component Object Model предусматривает два типа серверов: внутренние, представляющие собой исполняемые программы, и внешние, являющиеся динамически компонуемые библиотеками.

Приложение связывается с компонентом при помощи его имени. Для обеспечения отсутствия коллизий имен на разных машинах и в продуктах разных производителей COM-компоненты именуются при помощи 16-байтных чисел, вычисляемых специальной хэш-функцией. Эти номера носят название GUID – Globally Unique Identifier.

В силу соблюдения принципа инкапсуляции, реализация COM-компонента закрыта для стороннего пользователя (и, соответственно, приложения) – известен лишь поддерживаемый компонентом функционал. Доступ к этому функционалу осуществляется при помощи интерфейса, своеобразного контракта между пользователем и COM-объектом, который по сути представляет собой список указателей на реализованные функции. Подобно компонентам, COM-интерфейсы также имеют свой уникальный идентификатор – IID (Interface Identifier).

Связь между клиентом и сервером обеспечивает операционная система. В системном реестре Windows хранится информация о COM-сервере: его название, GIUD, уникальный номер компонента CLSID (Class Identifier).

Программное обеспечение «StabMed» реализует стандарт Component Object Model, выступая в роли COM-сервера. Посредством поддерживаемого интерфейса ISignalExporter сторонние приложения могут получать информацию о сигналах, импортировать сами сигналы, а также информацию о пациентах, в ходе обследования которых эти сигналы были получены. Полное описание реализации стандарта Component Object Model данным ПО приводится в Приложении 1.

В рамках данной работы при проектировании автоматизированной системы задача состоит в написании программного модуля, который бы обеспечивал программе доступ к нужным данным. Поскольку программная часть системы реализуется на языке Python, необходимо было найти реализацию компонентов Component Object Model на данном языке программирования. С этой целью была выбрана библиотека pywin32. Помимо наличия реализации всех требуемых для создания COM-компонентов функций, библиотека примечательна наличием подробного описания принципов и методов работы как в виде сопроводительной документации, так и в виде книги Python Programming on Win32 [6], материалы из которой и были использованы при реализации данного компонента автоматизированной системы.

В StabMed реализован внутренний COM-сервер, то есть сервер внутри EXE-приложения. Об этом говорит свойство InprocServer32 в описании соответствующей библиотеки типов, хранящемся в системном реестре Windows. Таким образом, в рамках взаимодействия с данным сервером необходимо зарегистрировать библиотеку типов – описание интерфейсов, поддерживаемых COM-сервером, которое хранится в системном реестре (рисунок )

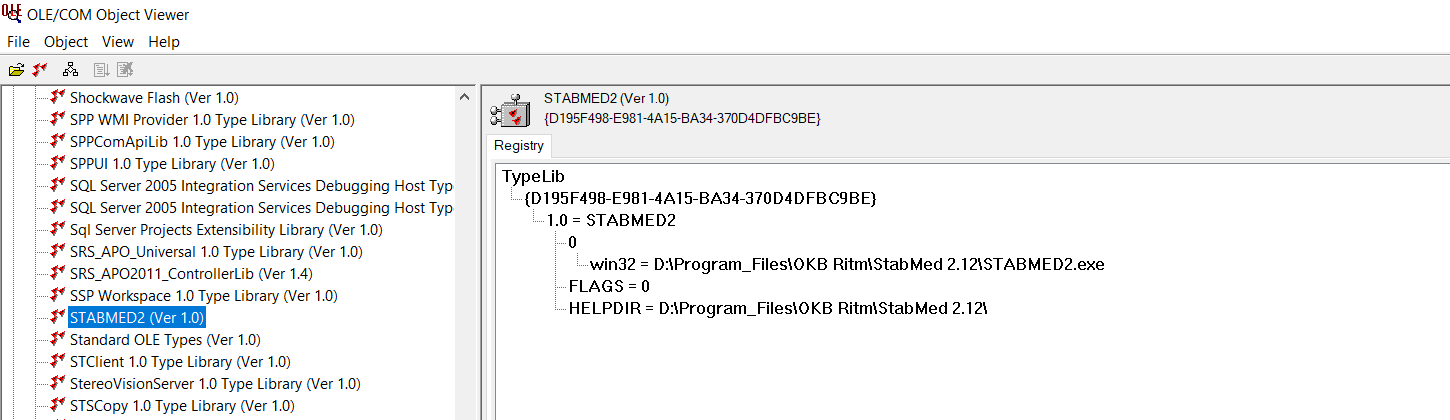


Рис. StabMed в числе других библиотек типов Component Object Model

Для этого необходимо выполнить скрипт makepy.py [6], поставляемый с вышеупомянутым пакетом pywin32, выбрав в открывшемся списке соответствующую библиотеку. После этого необходимые функции интерфейса становятся доступны Python (рис. )

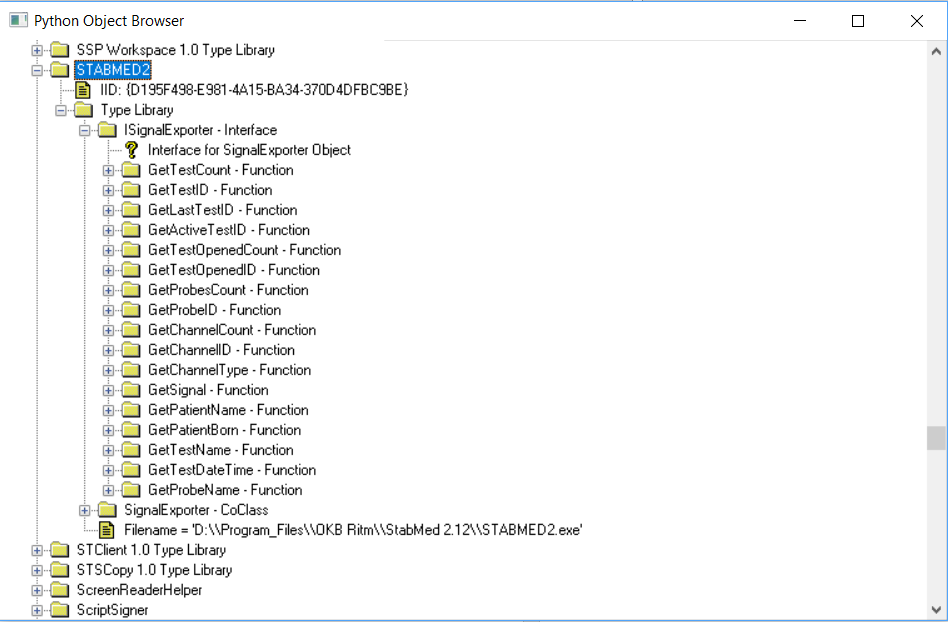


Рис. Браузер COM-объектов, доступных посредством Python

При этом автоматически генерируется описание библиотеки на языке IDL (Interface Definition Language – язык описания интерфейсов) со всеми необходимыми зависимостями. Теперь из любого скрипта на языке Python можно обращаться к COM-серверу при помощи метода win32com.client.Dispatch(), который возвращает Python-объект, имеющий все методы, описанные Приложении 1.

Суммируя все вышеизложенное, работу модуля проектируемой автоматизированной системы можно описать алгоритмом, блок-схема которого приведена на рисунке Реализация данного алгоритма с учетом всех требований стандарта Component Object Model приведена в Приложении 2.

Разработка графического пользовательского интерфейса

Для проектирования графического пользовательского интерфейса, соответствующего ранее выявленным требованиям, был язык Python 3.7 и

Разработка хранилища измерений

Для возможности дальнейшего использования полученных в процессе проведения стабилографического исследования данных, а также учитывая возможность в дальнейшем добавить в проектируемую автоматизированную систему инструменты анализа стабилограмм на базе методов машинного обучения, необходимо спроектировать хранилище получаемой системой информации.

Здесь стоит учесть, что методы стабилографии используются не только исследователями-физиологами, но и медицинскими работниками. То есть автоматизированная система по сути имеет дело с медицинскими данными. Для обеспечения удобства дальнейшего использования стабилометрических данных следует спроектировать хранилище в соответствии со стандартами хранения медицинской информации.

Таким образом, основных требований, влияющих на архитектуру, к проектируемому хранилищу данных два:

- ориентированность на дальнейшее использование методами машинного обучения;

- соответствие стандартам представления медицинской информации.

Для понимания сути первого требования необходимо понять, что представляет собой машинное обучение – таким образом, станет ясна суть требований, предъявляемых к используемым данным.

Машинное обучение (Machine Learning, ML) - это раздел информатики, изучающий алгоритмы и статистические модели, которые используются компьютерными системами для эффективного выполнения конкретной задачи без использования явных инструкций, вместо этого опираясь на шаблоны и собственные выводы. Оно рассматривается как раздел искусственного интеллекта. Алгоритмы машинного обучения строят математическую модель на основе выборочных данных, известных как «обучающие данные», для того, чтобы делать прогнозы или решения без явного программирования для выполнения задачи. Алгоритмы машинного обучения используются в широком спектре приложений, таких как фильтрация электронной почты и компьютерное зрение, где невозможно разработать алгоритм конкретных инструкций для выполнения задачи. Машинное обучение тесно связано с вычислительной статистикой, которая фокусируется на прогнозировании с использованием компьютеров. Важной частью машинного обучения является интеллектуальный анализ данных, который фокусируется на поисковом анализе данных. Применительно к бизнес-задачам машинное обучение также называется предиктивной аналитикой. Отличительной особенностью алгоритмов машинного обучения от обычных программных алгоритмов является то, что с каждой новой итерацией алгоритм выполняет возложенную на него функцию все лучше и лучше – он «учится».

В контексте пригодности сохраняемых данных для нужд машинного обучения следует рассмотреть существующие методы и алгоритмы и выявить некие общие требования, предъявляемые к исходным данным в этой сфере.

Поскольку машинное обучение является довольно обширной областью, следует определить группу задач, для решения которых будут применяться сохраняемые автоматизированной системой стабилометрические данные. На сегодняшний день можно выделить несколько проблем [14], решаемых методами машинного обучения (таблица )

|  |  |  |
| --- | --- | --- |
| Задача | Описание | Пример применения |
| Классификация | Для каждого объекта на основе данных определяется дискретный класс | Персонализация контента, выявление производственных дефектов, изучение эффективности лекарственных средств |
| Регрессия | На основе данных предсказывается фактическое значение параметра | Прогнозы на рынке ценных бумаг, управление рисками, прогноз погоды |
| Рекомендация | Предсказывается альтернатива, которую предпочтет пользователь | Предложения продуктов, подбор персонала, предложения контента |
| Заполнение пропусков | Вывод значений отсутствующих входных данных | Неполные истории болезни, данные переписей, неполная информация о клиентах |

Таблица Варианты применения машинного обучения с учителем, систематизированные по типам задач.

С другой стороны, если подходить к классификации методов машинного обучения с точки зрения принципов работы алгоритмов Machine Learning, то эти методы можно разделить на три большие группы:

- методы контролируемого обучения (обучение с учителем),

- методы неконтролируемого обучения,

- методы с частичным контролем.

Во каждой группе подразумевается поиск алгоритмом

На сегодняшний день существует ряд стандартов подобного рода. Таким образом, перед непосредственно проектированием и реализацией хранилища данных для автоматизированной системы следует выбрать стандарт, на котором будет базироваться хранилище. Ниже проводятся краткие обзор и сравнительный анализ различных стандартов, касающихся медицинских данных.

Поскольку задача унификации медицинской информации и информатизация в сфере