УДК 004.5

ИНТЕРФЕЙС СИСТЕМЫ «ОБРАБОТКА И АНАЛИЗ ИЗОБРАЖЕНИЙ И ВИДЕОПОТОКОВ»

¹Миссаров М.Д., ¹Шустова Е.П., ²Шустова К.П.

¹ФГАОУ ВПО «Казанский (Приволжский) федеральный университет», Казань,
e-mail: moukadas.missarov@kpfu.ru, evgeniyashustova@yandex.ru;

²ФГБОУ ВО «Казанский государственный энергетический университет», Казань,
e-mail: kseniyashustova@yandex.ru

В настоящей статье разработан интерфейс главного окна системы «Обработка и анализ изображений и видеопотоков». Он представлен на UML-диаграмме классов интерфейса с использованием основных идей шаблона проектирования Model-View-Controller. Наряду с известными принципами проектирования информационных систем, такими как системность, развитие, совместимость, стандартизация, унификация и эффективность, с помощью предложенного интерфейса реализуются также принципы обучаемости, многопрофильности, научности, открытости. Эта система позволяет пользователям тестировать классические, современные (опубликованные в научных журналах) и персональные методы обработки и анализа изображений и видеопотоков; сравнивать качество работы разных методов для решения одной задачи; использовать при анализе качества работы методов известные репозитории данных. Разрабатываемая система является как инструментом улучшения методов обработки и анализа изображений (видеопотоков), так и средством ознакомления с прикладными модулями, отчетами о качестве работы алгоритмов методов и прикладных модулей. Это может осуществляться с помощью применения базы данных, поставляемой с системой, с использованием своих материалов или с помощью общения через интернет-страницу системы с другими пользователями этой системы.

Ключевые слова: интерфейс системы, принципы проектирования, UML, диаграммы классов интерфейсов, обработка изображений, анализ изображений, прикладные модули, качество методов

INTERFACE OF SYSTEM «PROCESSING AND ANALYSIS OF IMAGES AND VIDEO STREAMS»

¹Missarov M.D., ¹Shustova E.P., ²Shustova K.P.

¹Kazan (Volga region) federal university, Kazan, e-mail: moukadas.missarov@kpfu.ru,
evgeniyashustova@yandex.ru;

²Kazan State Energy University, Kazan, e-mail: kseniyashustova@yandex.ru

In this article, the interface of the main window of the system «Processing and analysis of images and video streams» is developed. It is presented on the UML diagram of the interface classes using the basic ideas of the Model-View-Controller design pattern. Along with the well-known principles of designing information systems, such as: systemicity, development, compatibility, standardization and unification, efficiency, the following principles are also implemented using the proposed interface: learning ability, versatility, science, openness. This system allows users to: test classic, modern (published in scientific journals) and personal methods of processing and analysis of images and video streams; compare the quality of work of different methods to solve one problem; use well-known data repositories when analyzing the quality of work of methods. The system under development is both a tool for improving image processing and analysis methods (video streams), and a means of familiarizing oneself with application modules, reports on the quality of work of method algorithms and application modules. This can be done by using both the database supplied with the system and using its materials or by communicating through the system's web page with other users of this system.

Keywords: system interface, design principles, UML, interface class diagrams, image processing, image analysis, application modules, quality of methods

В настоящее время имеется большой объем информации о методах обработки и анализа изображений и видеопотоков. Но вся эта информация содержится в разных научных журналах, диссертациях и учебных пособиях в текстовом формате. Отметим, что в разработанных в настоящее время системах компьютерной алгебры и языках программирования (например, Python, C++, C#) есть возможность работать с методами обработки и анализа изображений. Но в них реализованы уже давно известные методы, причем в них

есть много недостатков: не отражены новые достижения в области методов обработки и анализа изображений и видеопотоков, что тормозит развитие научной мысли в этой области [1]. Отметим и другой аспект. Степень сотрудничества вузов России с фирмами – производителями информационных технологий как в области обработки и анализа изображений и видеопотоков, так и в области управления качеством такого анализа невысока.

Как показало изучение рынка существующих программ, на данный момент еще

нет системы, которая могла бы позволить пользователям:

- тестировать классические, современные (опубликованные в научных журналах) и персональные методы обработки и анализа изображений и видеопотоков (например, [2, 3]);
- сравнивать качество работы разных методов для решения одной задачи;
- использовать при анализе качества работы методов известные репозитории данных (например, [4]).

Поэтому создание такой системы носит инновационный характер. В настоящей работе разрабатывается интерфейс главного окна именно такой системы. Принципы проектирования и ее архитектура описаны в статье [1].

Разрабатываемая система является как инструментом улучшения методов обработки и анализа изображений (видеопотоков), так и средством ознакомления с прикладными модулями, отчетами о качестве работы алгоритмов методов и прикладных модулей. Это может осуществляться с помощью использования как базы данных (БД), предоставляемой с системой, так и с применением своих материалов или с помощью общения через интернет-стра-

ницу системы с другими пользователями этой системы.

Цель исследования: создание системы поддержки принятия решений, которая способствовала бы ускорению развития научной мысли в области обработки и анализа изображений и видеопотоков, а также разработке прикладных решений в этой области.

Объекты исследования: изображения, видеопотоки, методы обработки изображений, методы анализа изображений и видеопотоков.

*Интерфейс главного окна системы
«Обработка и анализ изображений
и видеопотоков»*

Рассмотрим принципы проектирования и архитектуру системы «Обработка и анализ изображений и видеопотоков», описанные в статье [1]. Ниже представим объекты, которыми надо управлять через интерфейс согласно этой архитектуре, и их взаимосвязи в виде схемы (рис. 1).

Поясним приведенную схему. Пользователь должен иметь возможность работать через систему управления интерфейсом с системой управления базой данных (СУБД) и системой управления базой моделей (СУБМ) следующим образом.

Рис. 1. Объекты ИС «Обработка и анализ изображений и видеопотоков»

Интерфейс для работы с СУБД должен содержать виджеты для извлечения информации из БД, поставляемой с системой, и обновления БД пользователя. В свою очередь обе эти БД должны иметь одинаковую архитектуру, а именно должны состоять из тестовых примеров, отчетов о качестве работы методов и прикладных модулей. Тестовые же примеры – это стационарные изображения и видеопотоки. Причем интерфейс должен позволять работать с видеопотоками как из файла, так и с камеры.

Интерфейс для работы с СУБМ должен содержать виджеты для извлечения информации из базы знаний (БЗ), поставляемой с системой, и обновления БЗ пользователя. В свою очередь обе эти БЗ должны иметь одинаковую архитектуру, а именно должны состоять из методов и справочной системы. Методы же подразделяются на методы тестирования качества работы алгоритмов и методы обработки и анализа (изображений и видеопотоков). Причем интерфейс СУБМ должен позволять работать с видеопотоками как из файла, так и с камеры и давать возможность работать как с классическими, так и с современными методами. В справочной системе должны быть приведены: теоретическое описание (метода, прикладного модуля), инструкция по работе с ИС и код программы для описываемого метода.

Реализуем проектирование интерфейса главного окна такой системы с использованием основных идей шаблона проектирования Model-View-Controller. Ниже приведем разработанную диаграмму классов этого интерфейса (рис. 2).

Поясним приведенную диаграмму. Диаграмма разработана с помощью ArgoUML. При ее разработке классы контроллера на схеме в силу возможности мы размещали друг относительно друга именно там, откуда на форме пользователь будет отправлять соответствующие команды. Поэтому блок «Контроллер» одновременно сообщает нам и дизайн главного окна. В блоке «Представление» расположение классов тоже соответствует порядку размещения соответствующих разделов. Согласно этой диаграмме главное окно системы (класс `cMainWindow`) разбиваем на левую и правую части. В левой части располагаем навигационную панель ИС (класс `cNavigationBar`), где пользователь выбирает нужный раздел, а в правой части (класс `cWorkSpace`):

- в верхней подобласти (класс `cMetaField`) слева отображается этот выбранный раздел (класс `cSelectedSection`), спра-

ва – всегда кнопка с разворачивающимся меню (класс `cUserMenu`) для настройки цвета (`FontColor`) и размера шрифта (`FontSize`) на формах пользователя, установки способа выделения (`metSelectedSection`) выбранного раздела (жирный шрифт, подчеркивание, подсветка и установка цвета подцветки), установки порядка отображения разделов на навигационной панели (`OrderSections`), а также вставки своего изображения на эту кнопку (`Download`) (например, своего фото) и текста слева от нее (`NameEntry`) (например, своего имени);

- в нижней части подобласти отображается сама выбранная форма (класс `cSelectedForm`).

Здесь везде буква «с» в начале наименования класса означает, что это класс контроллера.

Из диаграммы видим, что класс представлений `View_IS` наследует методы двух классов: класса представления меню пользователя `vUserMenu` и класса представления навигационной панели `vNavigationBar`.

Класс `vUserMenu` наследует свойства классов `vUserMapping` и `vColorFontManagement`, отвечающих соответственно за:

- отображение пользователя на кнопке меню и его имени в правом верхнем углу формы главного окна;

- управление цветом форм и шрифтом во всех окнах системы.

Это означает, что в интерфейсе в правом верхнем углу пользователь будет видеть кнопку с разворачивающимся меню. В этом разворачивающемся меню он увидит две строки:

- отображение пользователя (класс `vUserMapping`);

- управление цветом форм и шрифтом (класс `vColorFontManagement`).

Из блока представления навигационной панели `vNavigationBar` видим, что на навигационной панели пользователь может выбрать разделы:

- Личный кабинет пользователя (класс `vUserAccount`);

- Методы (класс `vMethods`);

- Репозиторий данных (класс `vDataRepository`);

- Тестирование работы методов (класс `vMethodTesting`);

- Отчеты о качестве работы методов (Reports on the quality of the working of methods) (класс `vReportsQuality`);

- Прикладные модули (Application modules) (класс `vApplicationModules`);

- Справка (класс `vHelp`).

Каждый из этих разделов пользователь может выбрать (метод `SelectedSection`), развернуть и свернуть (метод `ExpandCurtail`).

Рис. 2. Диаграмма классов интерфейса главного окна системы
«Обработка и анализ изображений и видеопотоков»

Раздел «Личный кабинет пользователя» можно развернуть на два подраздела:
– профиль пользователя (UserProfile);
– другие пользователи ИС (OtherUsers).

Подраздел «Профиль пользователя» в свою очередь можно развернуть на два подраздела:
– о себе (класс AboutMe);

– портфель (Portfolio).

Каждому подразделу раздела UserAccount соответствуют два ресурса: интернет-страница и форма в рамках ИС, которые формируются при срабатывании методов moUASite и moFormUA из класса moUserAccount соответственно. Все изменения, которые внесены в «Личный кабинет пользователя» в рамках локальной ИС, считываются соответственно и на интернет-страницу, если пользователь установил опцию: отобразить на сайте (см. метод moUASite).

В рамках локальной ИС на форме «Профиль пользователя» пользователь:

- вводит краткую информацию о себе (имя, e-mail, области интересов);
- открывает или закрывает доступ другим пользователям системы к материалам своего портфеля;
- фильтрует информацию своего портфеля согласно его запросу [1].

Портфель пользователя – его собственные файлы изображений и видеопотоков, отчетов о качестве работы методов, собственные методы и прикладные модули [1]. Информация о виде открытости материала портфеля синхронно отображается и на всех других формах ИС. Если эта информация будет изменена непосредственно на этих других формах, то она так же синхронно изменится и в «Портфеле пользователя»;

- система отображает открытые для использования другим пользователям материалы владельца личного кабинета с указанием их вида открытости: для всех или по требованию.

Подразделу «Другие пользователи ИС» соответствует форма, на которой отображены другие пользователи системы, их материалы, блок общения с конкретным пользователем, форум.

Раздел «Методы» можно развернуть на три подраздела:

- методы обработки изображений и видеопотоков (класс vProcessingMethods);
- методы анализа изображений и видеопотоков (класс vAnalysisMethods);
- методы тестирования качества работы алгоритма метода (класс vQualityTestingMethods).

Раздел «Репозиторий данных» можно развернуть на три подраздела:

- видеопотоки (класс vVideoStreams);
- стационарные изображения (класс vStationaryImages);
- прикладные модули (класс vApplicationModules).

Раздел «Тестирование работы методов» можно развернуть на три подраздела:

- мои тестирования (класс vMyTesting);
- тестирования, поставляемые с системой (tests supplied with the system) (класс vSystemTests);

- пользовательские тестирования (класс vCustomTestings).

Раздел «Отчеты о качестве работы методов» можно развернуть на подразделы:

- мои отчеты (класс vMyReports);
- отчеты, поставляемые с системой (reports supplied with the system) (класс SystemReports);
- отчеты других пользователей (reports of other users) (класс vReportsOtherUsers).

Раздел «Прикладные модули» можно развернуть на подразделы:

- мои прикладные модули (класс vMyAppModules);
- прикладные модули, поставляемые с системой (класс vSystemAppModules);
- прикладные модули других пользователей (класс vOtherAppModules).

Раздел «Справка» можно развернуть на подразделы:

- инструкция по работе с ИС (класс vInstrWIS);
- справка по методам (класс HelpMethods).

Выбранный пользователем раздел по умолчанию выделяется системой жирным шрифтом.

Контроллер (Controller) – классы, воспринимающие команды пользователя. Он интерпретирует действия пользователя, оповещая модель о необходимости изменений. Класс главного окна cMainWindow полностью владеет остальными классами. Он отвечает за весь их жизненный цикл, включая создание и уничтожение. Поэтому тип отношений – композиция.

Контроллер принимает команды пользователя и в соответствии с ними изменяет состояние модели. Всякий раз, когда состояние модели изменяется, модель шлет об этом уведомление представлению. Получив уведомление о том, что состояние модели изменилось, представление обновляет визуализацию, перерисовывая ее в соответствии с состоянием модели. И, наконец, пользователь, видя эти изменения, воспринимает их как реакцию приложения на свои действия. При этом сама модель ничего не знает ни о представлениях, которые ее визуализируют, ни об используемых этими представлениями способах визуализации.

У нас каждое представление имеет ссылку на модель и само решает, какие части модели и каким образом визуализировать. Это показано на диаграмме при помощи ссылки на модель из базового класса представлений View_IS в класс model_IS. Каждое представление зависит от модели.

Рис. 3. Интерфейс главного окна системы «Обработка и анализ изображений и видеопотоков» с выбранным разделом

При клике на подразделы срабатывают соответствующие методы из модели и открывается соответствующая форма. Методы из модели помечены на диаграмме символами «mo» в начале наименования метода, а соответствующие им методы представления – символом «v». Все формы, кроме форм для меню пользователя и раздела «Справка», открываются в нижней части рабочей области, а пока они еще не открылись, на этом месте вращается кружок, который сигнализирует о том, что запрос принят и идет обработка (см. метод `status_message1`). Формы для меню пользователя и раздела «Справка» открываются в своих отдельных окнах. Состояние системы при клике на раздел «Справка» или на меню пользователя отмечается вращающимся кружком около них (см. метод `status_message2`).

Заметим, что мы поместили всю функциональность контроллера, отвечающую за выдачу наименования (метод `name_requested_form`) и самой формы выбранного раздела навигационной панели (метод `form_request`), в один класс срабатывания `moOutput`. Мы сделали это с целью избежать дублирования кода для этих операций.

Заметим, что система разрабатывается для русскоязычного пользователя. Ниже представим соответствующий этой диаграмме примерный вид формы главного окна с выбранным пользователем разделом «Тестирование работы методов» на навигационной панели (рис. 3). Интерфейс формы этого раздела описан в работе А.Ю. Соловьяненко [5] и встроен в эту систему.

Заключение

Наряду с известными принципами проектирования информационных систем, такими как системность, развитие, совместимость, стандартизация, унификация и эффективность, с помощью предложенного интерфейса реализуются также принципы обучаемости, многопрофильности, научности, открытости. Заметим, что разработанный нами интерфейс позволяет реализовать принцип открытости системы двумя способами: без выхода в интернет и с выходом в него, а именно: без выхода в интернет можно использовать модули для добавления (удаления, обновления) пользовательских баз тестовых примеров изображений и видеопотоков, методов, прикладных модулей и отчетов о качестве исполнения алгоритмов; с выходом на сайт этой системы – работать в своем личном кабинете. В своем профиле пользователь увидит новости и других пользователей этой системы, если они в своем личном кабинете разрешили видимость себя и своих сообщений на этой странице. Эта информация может быть использована для общения и обмена материалами между пользователями. Доступ к своим материалам пользователь открывает по своему усмотрению и на тех условиях, которые будут обговорены вне нашей системы.

Пользователь может получать (давать) доступ к пользовательским (соответственно своим) разработанным методам обработки или анализа изображений и видеопотоков, методам тестирования алгоритмов, при-

кладным модулям, отчетам по результатам тестирования, а также тестовым изображениям или видео. Прикладные модули могут быть использованы также и как самостоятельные системы.

Разрабатываемая система внесет весомый вклад в обеспечение интеграции образования, науки и производства, что является залогом успешного выполнения управления качеством и, как следствие, обеспечения конкурентоспособности отечественных систем интеллектуального видеоанализа. В этой системе отражены новые достижения в области методов обработки и анализа изображений и видеопотоков (со ссылками на современные научные статьи и репозитории), что будет способствовать ускорению развития научной мысли в этой области.

Список литературы

1. Shustova E.P. Design principles and architecture of the system «Processing and analysis of images and video streams». Journal of Physics: Conference Series. 2019. Vol. 1158(4):042017. DOI: 10.1088/1742-6596/1158/4/042017.
2. Белим С.В., Ларионов С.Б. Алгоритм сегментации изображений, основанный на поиске сообществ на графах // Компьютерная оптика. 2016. Т. 40. № 6. С. 904–910. DOI: 10.18287/2412-6179-2016-40-6-904-910.
3. Шаталин Р.А., Фидельман В.Р., Овчинников П.Е. Обнаружение нехарактерного поведения в задачах видеонаблюдения // Компьютерная оптика. 2017. Т. 41. № 1. С. 37–45. DOI: 10.18287/2412-6179-2017-41-1-37-45.
4. Princeton Tracking Benchmark. [Электронный ресурс]. URL: <http://tracking.cs.princeton.edu/dataset.html> (дата обращения: 20.05.2020).
5. Соловяненко А.Ю. Система тестирования качества работы методов обработки и анализа изображений и видеопотоков // XXIV Туполевские чтения: материалы Международной молодежной научной конференции (Казань, 7–8 ноября 2019 г.). Казань: Изд. ИП Сагиева А.Р., 2019. Т. 4. С. 417–423.