

УДК 004.4

В.А. Емельянов

Донбасский государственный технический университет, Алчевск, Украина

ОБЪЕКТНАЯ МОДЕЛЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ПРОЦЕССА МЕТАЛЛОГРАФИЧЕСКОГО КОНТРОЛЯ КАЧЕСТВА МЕТАЛЛОВ

Показана актуальность создания программного обеспечения для процесса металлографического анализа. Приведена объектная модель программного обеспечения системы металлографического контроля качества металлов, построенная с помощью унифицированного языка моделирования (UML). Модель отображает основные абстракции предметной области, варианты использования программного обеспечения и организацию программных модулей. Объектная модель является основой для построения специализированного программного обеспечения.

Ключевые слова: объектная модель программного обеспечения, металлографический анализ, унифицированный язык моделирования (UML).

Введение

В целях повышения качества изделий предприятия постоянно увеличивают объемы операций контроля и численность контролирующего персонала. Важным средством решения проблемы контроля качества на предприятии является применение объективных физических методов контроля, таких как металлографический анализ [1]. Постоянное повышение требований, предъявляемых к качеству черных металлов разных групп и классов, вызывает необходимость разработки моделей и средств автоматизированного контроля качества продукции.

Как известно [1, 2], в основу металлографического анализа положена интерпретация изображений микроструктур металлов. Уменьшение времени, затрачиваемого для данной операции, представляется возможным с помощью программного обеспечения, которое обеспечит технолога возможностью автоматической обработки изображений микроструктур металлов. Поэтому актуальной становится задача разработки специализированного программного обеспечения для процесса металлографического анализа, которое будет осуществлять анализ изображений микроструктур металлов.

Разработка объектной модели программного обеспечения для металлографического анализа

На современном этапе развития информационных систем и технологий структуре и организации программного обеспечения уделяется особое внимание.

При разработке структуры программного обеспечения был использован объектно-ориентированный подход [3]. В основе объектно-ориентированного подхода лежит объектная декомпозиция, т.е. представление разрабатываемого программного

обеспечения в виде совокупностей объектов, в процессе взаимодействия которых, через передачу сообщений, происходит выполнение необходимых функций [4].

Спецификация разрабатываемого программного обеспечения объединяет в себе следующие модели:

1. Модель использования – представляет собой описание функциональности программного обеспечения с точки зрения пользователя.

2. Концептуальная модель – модель, описывающая основные абстракции предметной области, которые обеспечивают требуемую функциональность ПО и их взаимодействие.

3. Модель реализации – определяет реальную организацию программных модулей и файлов.

Для построения этих моделей был использован унифицированный язык моделирования (UML).

В модели использования проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью так называемых вариантов использования. Вариант использования служит для описания сервисов, которые система предоставляет актеру.

При этом модель не отображает, каким образом будет реализовано взаимодействие актеров с системой. Модель использования приведена на рис. 1.

В модели выделены 6 вариантов использования и 1 актер (технолог), между которыми установлены отношения включения и расширения. Значения указанных в модели кратностей отражают общие правила обработки изображений микроструктур металлов.

Согласно этим правилам, один технолог может обрабатывать множество изображений. В модели вариант использования «Провести анализ» уточнен на основе введения еще трех дополнительных вариантов использования.



Рис. 1. Модель использования

Это следует из более детального анализа самого процесса, что позволяет выделить в качестве отдельных сервисов такие действия, как распознавание изображения, определение характеристик по распознанному изображению и генерирование экспертного вывода об исследуемом образце. Указанные действия раскрывают поведение исходного варианта использования в смысле его конкретизации, и поэтому между ними будет иметь место отношение включения.

С другой стороны, проведение анализа предполагает наличие самостоятельного информационного объекта – нейронной сети. В нашем случае, нейронная сеть может обучаться технологом на уже существующем наборе изображений микроструктур металлов.

Таким образом, на основании описания данной модели представляется возможным построить концептуальную модель ПО. Концептуальная модель ПО приведена на рис. 2.

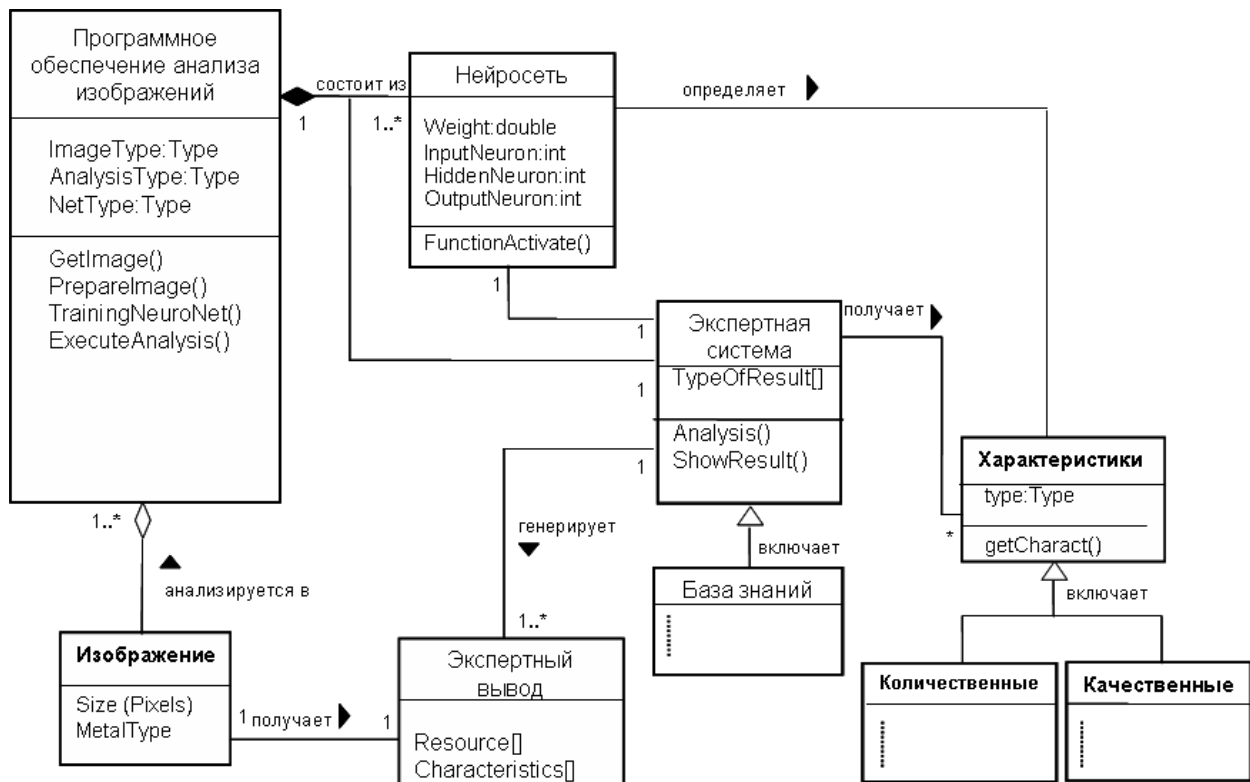


Рис. 2. Концептуальная модель программного обеспечения анализа изображений микроструктур металлов

Модель представлена в виде совокупности классов. Для данной модели характерно отношение «целое – часть». В модели можно выделить класс «Программное обеспечение» как «целое», а все остальные классы являются его «частями».

В модели существуют такие классы как: «Изображение», «Экспертный вывод», «Нейронная сеть», «Характеристики». Между классами «Изображение» и «Экспертный вывод» есть ассоциация, показывающая, что для изображения микроструктуры металла может существовать экспертный вывод, отражающий информацию количественного и качественного характера. Более того, для каждого конкретного изображения микроструктуры предназначена своя экспертная оценка, что отражено в виде связи «один к одному». Классы «Экспертная система» и «экспертный вывод» соединены отношением «один ко многим», т.е. одна экспертная система может генерировать множество экспертных выводов для множества изображений металлов.

Классы «Программное обеспечение», «Нейронная сеть» и «Экспертная система» содержат несколько операций, позволяющих манипулировать их частями. Эти операции включены из-за их важности для поддержания целостности данных (например, изменение типа нейронной сети в классе «Программное обеспечение» затрагивает другие классы, такие как «Нейронная сеть»).

Программное обеспечение может анализировать любое количество изображений микроструктур металлов, что отражено соответствующей связью «1..*». Программное обеспечение может состоять из

одной или нескольких нейронных сетей и только из одной экспертной системы.

Отношения между классом «Программное обеспечение» и классами «Изображение» и «Нейронная сеть» отличаются друг от друга, хотя оба являются отношениями агрегирования. Отношение между классом «Программное обеспечение» и классами «Нейронная сеть», «Экспертная система» является композитным агрегированием, т.к. эти классы являются неотъемлемой частью целого – класса «Программное обеспечение», а класс «Изображение» является отдельной частью.

Класс «Характеристики» включает в себя количественные и качественные характеристики распознанного изображения микроструктуры металла, что отражено в виде операции включения соответствующих классов.

Созданные ранее модели отражали концептуальные аспекты построения объектной модели системы и относились к логическому уровню представления. Основное назначение логического представления состоит в анализе структурных и функциональных отношений между элементами модели системы. Однако для создания конкретной физической системы необходимо некоторым образом реализовать все элементы логического представления в конкретные материальные сущности. Для описания таких реальных сущностей предназначен другой аспект модельного представления, а именно физическое представление объектной модели. Для физического представления построена модель реализации, представленная на рис. 3:

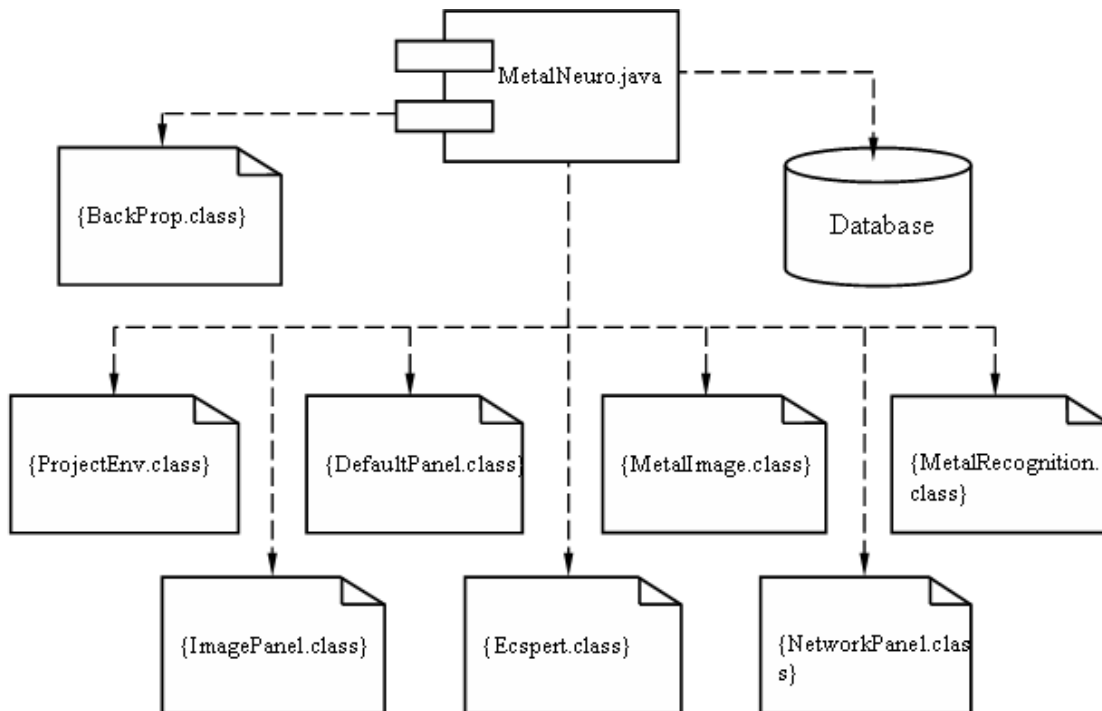


Рис. 3. Модель реализации ПО

В модели описаны особенности физического представления системы, установлены зависимости между программными компонентами.

Компонент MetalNeuro.java реализует определенный набор интерфейсов для взаимосвязи с базой данных и модулями, а также служит для общего обозначения элементов физического представления модели.

Модуль класса BackProp.class реализует свойства и методы нейронной сети, а также алгоритм обратного распространения ошибки.

Модуль класса ProjectEnv.class содержит информацию о текущем состоянии системы.

Модуль класса DefaultPanel.class – описывает основную панель графического интерфейса пользователя (Graphic User Interface).

Модуль класса MetalImage.class – реализует свойства и методы для доступа к текущему обрабатываемому изображению.

Модуль класса MetalRecognition.class связывает все модули, обеспечивая функционирование информационных потоков.

Модуль класса ImagePanel.class содержит методы класса для визуального представления результатов распознавания.

Модуль класса Ecsper.class содержит методы, свойства и функции необходимые для экспертной системы.

Модуль класса NetworkPanel.class – содержит класс, который наследует методы класса для визуальной работы с нейронной сетью.

Выводы

Таким образом, построенная объектная модель отображает основные абстракции предметной области, варианты использования программного

обеспечения, его физическое представление, а также информационные потоки, функционирующие в программном обеспечении, необходимом для процесса металлографического анализа.

На основании предложенной объектной модели было разработано программное обеспечение [5], позволяющее проводить автоматизированную обработку изображений микроструктур металлов, которое также позволяет проводить поддержку принятия решений относительно исследуемого образца металла.

Список литературы

1. Богомолова Н.А. Практическая металлография / Н.А. Богомолова. – М.: Высшая школа, 1987. – 240 с.
2. Шульте Ю.А.. Комплексный контроль качества конструкционной стали / Ю.А. Шульте. – К.: Техника, 1986. – 125 с.
3. Грэхем И. Объектно-ориентированные методы. Принципы и практика / И. Грэхем. – М.: Вильямс, 2004. – 768 с.
4. Якобсон И. Унифицированный процесс разработки программного обеспечения / И. Якобсон, Г. Буч, Дж. Рамбо. – СПб.: Питер, 2002. – 458 с.
5. Емельянов В.А. Программное обеспечение автоматизированной компьютерной системы металлографического контроля качества металлов / В.А. Емельянов // Труды 11-й международной научно-технической конференции «Интегрированные компьютерные технологии в машиностроении – 2009». – Т. 2. – X., 2009. – С. 4.

Поступила в редколлегию 8.12.2009

Рецензент: д-р техн. наук, проф. В.С. Харченко, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.

ОБ'ЄКТНА МОДЕЛЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПРОЦЕСУ МЕТАЛОГРАФІЧНОГО КОНТРОЛЮ ЯКОСТІ МЕТАЛІВ

В.О. Емельянов

Показана актуальність створення програмного забезпечення для процесу металографічного аналізу. Приведена об'єктна модель програмного забезпечення системи металографічного контролю якості металів, яка побудована за допомогою уніфікованої мови моделювання (UML). Модель відображає основні абстракції наочної області, варіанти використання програмного забезпечення і організацію програмних модулів. Об'єктна модель є основою для побудови спеціалізованого програмного забезпечення.

Ключові слова: об'єктна модель програмного забезпечення, металографічний аналіз, уніфікована мова моделювання (UML).

OBJECT MODEL OF SOFTWARE FOR PROCESS OF METALLOGRAPHIC CHECKING OF METALS QUALITY

V.A. Yemelyanov

Actuality of creation of software is presented for the process of metallographic analysis. The object model of software of metallographic system checking of metals quality is resulted, which developed by a Unified Modeling Language (UML). The model represents basic abstractions of subject domain, variants of the use of software and organization of the program modules. The object model is basis for the creating of the specialized software.

Keywords: object model of software, metallographic analysis, Unified Modeling Language (UML).