

оптимизация данного ПО, и адаптация алгоритмов реконструкции для их использования на современных мобильных платформах.

Литература

1. Гонсалес Р. Цифровая обработка изображений в MATLAB/Р. Гонсалес, Р. Вудс, С. Эддинс. – М.: Техносфера, 2006. – 616 с.
2. Гонсалес Р. Цифровая обработка изображений/Р. Гонсалес, Р. Вудс. – М.: Техносфера, 2005. – 1072 с.

Анотація

У роботі розглядається алгоритм відновлення спотворених зображень в умовах неповної апріорної інформації. Ефективна реалізація цієї задачі є актуальною проблемою, тому в роботі пропонується реалізація алгоритму відтворення зображень на основі фільтру Вінера. Подальшим розвитком роботи є адаптація алгоритму до використання на мобільних платформах.

Ключові слова: реконструкція зображень, перетворення Фур'є, згортка, оператор спотворення, деконволюція.

Аннотация

В работе рассматривается алгоритм восстановления искаженных изображений в условиях неполной априорной информации. Эффективная реализация этой задачи является актуальной проблемой, поэтому в работе предлагается реализация алгоритма восстановления изображений на основе фильтра Винера. Дальнейшим развитием работы является адаптация алгоритма к использованию на мобильных платформах.

Ключевые слова: реконструкция изображений, преобразование Фурье, свертка, оператор искажения, деконволюция.

Abstract

We consider the distorted image restoration algorithm in conditions of incomplete prior information in this work. Effective implementation of this task is an urgent problem, so we propose the implementation of image restoration algorithm based on the Wiener filter. Adaptation algorithm for use on mobile platforms is further development work.

Keywords: image restoration, Fourier transform, convolution, distortion operator, deconvolution.

АВТОМАТИЧЕСКОЕ ПОСТРОЕНИЕ UML-ДИАГРАММ В ПРОЦЕССЕ СОПРОВОЖДЕНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Бабаков Р.М., канд. техн. наук, доцент, cp1d@mail.ru;

Зиновьева Е.Б., студент, lizochka313@gmail.com

ДонНТУ, Красноармейск, Украина

В настоящее время производство ПО является отдельной, достаточно широкой отраслью индустрии, к которой задействовано множество специалистов в различных странах. Сегодня ПО является неотъемлемой составной частью многих промышленных продуктов, является основой много-

численных сервисов и технологий в бизнесе, образовании, в социальной инфраструктуре и бытовой жизни. В программную индустрию вовлечены большие денежные средства. Рост производства в этой области продолжается высокими темпами.

При этом разработка ПО остается достаточно рискованной деятельностью с точки зрения вложения инвестиций. Низка предсказуемость ресурсов и времени разработки проектов, существует много проблем с соответствием созданного ПО требованиям контекста, где оно будет работать, а также ожиданиям заказчика. Высок процент неудачных проектов по сравнению с другими промышленными областями. Когда разработка ПО является составной частью более крупного промышленного проекта (например, создается новая модель автомобиля, разрабатывается космический корабль), то оказывается, что программная часть разработки является наиболее дорогостоящей и плохо предсказуемой. [1]

Накопленный к настоящему времени опыт создания программных систем показывает, что это сложная и трудоемкая работа, требующая высокой квалификации участвующих в ней специалистов. Однако до настоящего времени создание таких систем нередко выполняется на интуитивном уровне с применением неформализованных методов, основанных на искусстве, практическом опыте, экспертных оценках и дорогостоящих экспериментальных проверках качества функционирования ПО. По данным Института программной инженерии (Software Engineering Institute, SEI) в последние годы до 80% всего эксплуатируемого ПО разрабатывалось вообще без использования какой-либо дисциплины проектирования, методом «code and fix» (кодирования и исправления ошибок). [2]

Сопровождение (maintenance) ПО — это процесс сбора информации о качестве ПО в эксплуатации, устранения обнаруженных в нем ошибок, его доработки и модификации, а также извещения пользователей о внесенных в него изменениях. [3] Качественное сопровождение программных продуктов невозможно без использования достаточного документирования. В процессе составления документации участвуют в той или иной мере все члены коллектива разработчиков. Для качественного выполнения процессов сопровождения и поддержки продукта часто критически необходимо наличие соответствующей документации к программному продукту. Часто можно услышать заявления о том, что код является самодокументируемым. Однако большое количество комментариев это не одно и то же, что и наличие качественной документации.

В своей практике руководители проектов довольно часто сталкиваются с тем, что программисты младшего звена не в состоянии подготовить достаточную документацию к своему коду. Чаще всего с этим сталкиваются профессионалы, которые пишут код интуитивно. В случае, если дать понять, что без документации никуда не деться, то вероятнее всего доку-

ментация будет подготовлена, однако не гарантируется качество документов.

При подготовке документации сложно и трудозатратно обходиться одним только текстом. Много времени может уйти на описание каких-либо данных, которые можно представить графически. А поскольку большую часть времени программисты работают со структурированными данными, то лучшим способом представлять эти данные являются диаграммы и схемы различного рода.

Представление данных в графическом виде позволяет решать самые разнообразные задачи. Наглядность является основным достоинством такого представления.

Многие авторы научных и обзорных статей, посвященных моделированию данных и объектов, в том числе систем реального времени, отмечают удобство UML относительно других нотаций. В числе прочих преимуществ языка UML следует отметить его достаточность для описания практически любой задачи и его достаточная простота и распространенность, что позволяет значительно сократить затраты на обучение сотрудников.

Однако, следует отметить, что UML появился и приобрел свою популярность довольно давно, а проблема с программистами, которые не всегда способны составить достаточную документацию все еще актуальна. Необходимо провести повышение эффективности использования UML-диаграмм при составлении документации к программному обеспечению.

Программисты большую часть времени работают со строго структурированным текстом. В случае предоставления возможности описания UML-диаграмм систем структурированным языком позволит значительно повысить эффективность использования диаграмм и улучшить качество документации к коду.

Разумеется, уже в настоящее время разработаны средства визуального программирования на основе UML, обеспечивающие интеграцию, включая прямую и обратную генерацию кода программ, с наиболее распространенными языками и средами программирования. И в ряде случаев такая методика создания диаграмм покажет себя более эффективной для создания диаграмм. Однако, для генерации диаграмм с кода необходимо предоставить сам код, что невозможно на начальных этапах разработки, а также на основании кода возможна генерация только определенных видов диаграмм, что не в полной мере отображает функциональность программного продукта.

Таким образом, создание программного обеспечения, которое будет преобразовывать структурированный текст в графическую информацию и его сочетание с существующими методами построения UML-диаграмм, должно существенно облегчить задачу разработчиков при создании документации к коду и соответственно сам процесс сопровождения ПО.

Литература

1. Кознов Д. В. Основы визуального моделирования / Д. В. Кознов. — М. : Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2008. — 246 с.: ил. — (Серия «Основы информационных технологий»).
2. Вендров А. М. Современные технологии создания программного обеспечения / А.М. Вендров. // Jet Info Online. — 2004. — № 4. — Режим доступа: URL: <http://www.jetinfo.ru/searchresult/?searchid=2202569&text=Методические%20основы%20технологий%20создания%20ПО&web=0>. — Название с экрана.
3. Зубкова Т. М. Технология разработки программного обеспечения: Учебное пособие / Т. М. Зубкова — Оренбург: ГОУ ОГУ, 2004. — 101 с.

Анотація

Процес супроводу коду може бути значною мірою полегшено за рахунок візуального представлення програмного забезпечення. Графіка є важливим аспектом при описі програмного продукту. Проте, багато програмісти нехтують необхідністю опису коду з використанням графічного інструментарію. Таким чином, створення програмного забезпечення, який спрощує створення графічних моделей програмного забезпечення, шляхом перетворення структурованого тексту в графічну інформацію, представляється дуже актуальним.

Ключові слова: UML, діаграми, структурований текст.

Аннотация

Процесс сопровождения кода может быть в значительной мере облегчен за счет визуального представления программного обеспечения. Графика является важным аспектом при описании программного продукта. Однако, многие программисты пренебрегают необходимостью описания кода с использованием графического инструментария. Таким образом, создание программного обеспечения, упрощающего создание графических моделей программного обеспечения, путем преобразования структурированного текста в графическую информацию, представляется очень актуальным.

Ключевые слова: UML, диаграммы, структурированный текст.

Abstract

Process tracking code can be greatly facilitated by a visual representation of the software. Graphics is an important aspect in the description of the product. However, many programmers neglect the need to describe the code using graphical tools. Thus, the creation of software that simplifies the creation of graphic models of software, by converting structured text in the graphic information is very important.

Keywords: UML, diagram, structured text.