

Для возможности использования двойной обработки характеристики (сравнения с эталонной характеристикой и с базой прецедентов) необходимо:

- ввести величину максимального отклонения текущей характеристики от эталонной, что является признаком начала процесса сравнения характеристики с базой прецедентов;
- ввести признак максимального отклонения от наиболее близкого прецедента для выдачи заключения о том, что данная ситуация в прошлом не возникала.

Также в данной системе необходимо реализовать возможность изучения новых проблемных ситуаций и занесений их в базу прецедентов.

Реализация дополнительного модуля вывода на основе прецедентов в СППР оценки технического состояния механизмов по вибрационным характеристикам пуска-выбега позволит повысить эффективность работы системы за счет увеличения количества методов анализа характеристики.

Литература

1. Aamodt A., Plaza E. Case-based reasoning: foundational issues, methodological variations, and system approaches // AI Communications. IOS Press. Vol. 7: 1. 1994. – P. 39-59.
2. Бранцевич П.Ю., Гончарова С.А. Программная система обработки и анализа вибрационных характеристик выбега/ «Известия Белорусской инженерной академии» №1(15)/2 – Минск, 2003. - с. 26-28.

РАЗРАБОТКА ПАКЕТА ПРИКЛАДНЫХ ПРОГРАММ СИНТЕЗА СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ НА БАЗЕ UML

Жуляк Н. А.

Белорусский государственный технологический университет, г. Минск

Одной из важнейших задач при разработке системы становится уменьшение количества ошибок, вносимых в нее. При этом значительно возрастает роль стадий проекта, предшествующих его непосредственной реализации.

Кризис разработки сложного программного обеспечения, а особенно корпоративных информационных систем (КИС), как правило, выражается в следующих обстоятельствах: проекты не сдаются в срок; существенно перерасходуется бюджет проектов; проекты не удовлетворяют заданным спецификациям; модификация проектов становится чрезвычайно трудоемкой и рискованной.

Преодоление этого кризиса связывают с объектно-ориентированной технологией программирования. Теперь говорят о начале "революции в программировании". При этом имеется в виду уже не объектно-ориентированное программирование, а разработка проектов КИС на основе визуального моделирования компонент (CBD – Component Based Development).

Современные инструментальные средства визуальной CBD-разработки характеризуются следующими свойствами:

1. Поддержка генерации кода и обратного проектирования (т.е. восстановление визуальной модели по программному коду) сразу для нескольких языков, включая: Visual Basic, C++, Java, PowerBuilder, CORBA Interface Definition Language (IDL), а также Data Definition Language для большинства СУБД [1].

2. Поддержка визуального объектно-ориентированного моделирования и полная совместимость с языком UML (Unified Modeling Language), который, начиная с 1997 года, задает стандарт для языков описания моделей.

В настоящее время широко известны два подхода к проведению декомпозиции системы при построении ее визуальной модели:

1. Основанная на упорядочении событий/потоков данных алгоритмическая декомпозиция (структурный анализ/проектирование).

2. Объектно-ориентированная декомпозиция, базирующаяся на выделении агентов, которые или сами действуют, или являются объектами действия.

Объектно-ориентированное программирование – это метод реализации, в результате применения которого составляются программы, организованные как совокупности взаимодействующих друг с другом объектов. Каждый объект является экземпляром некоторого класса, а все классы являются членами иерархии классов, объединенной отношением наследования. Объектно-ориентированное проектирование состоит в объектной декомпозиции системы. Для визуального выражения разных аспектов модели, логического (структура классов и объектов) и физического (модульная и процессорная архитектура), используется различная нотация.

Визуальное моделирование на UML представляет собой единый подход, который используется для итерационного "изобретения" решения имеющейся проблемы [2]. Это обеспечивает "движение модели" от требований заказчика к программной реализации. Цель визуального моделирования КИС состоит в том, чтобы полученная информационная система удовлетворяла следующим условиям: соответствие заданным (возможно, неформальным) спецификациям; соответствие ограничениям целевой аппаратной платформы; соответствие (явным и/или неявным) требованиям к производительности при заданных ограничениях на аппаратные ресурсы; соответствие требованиям, предъявленным к процессу разработки: стоимость разработки, сроки и т.п.

Методология визуального моделирования помогает преодолеть типичные организационные трудности, возникающие перед командами разработчиков, группами поддержки качества и менеджерами, участвующими в создании сложного программного обеспечения масштаба предприятия, а именно: перерасход бюджета, несоблюдение этапных сроков и проблемы качества продукта.

Реально процесс разработки состоит из нескольких стадий. Причем эти стадии существуют в любых моделях жизненного цикла проектов, они лишь отличаются названиями и группировкой действий.

Цели стадии анализа требований состоят в том, чтобы понять процессы, которые управляют предприятием или системой, определить область деятельности системы и требования пользователя. Система рассматривается с точки зрения конечного пользователя как «черный ящик», составляется представление, что система будет делать, не рассматривая, как она это будет делать.

На данной стадии с помощью UML создается модель прецедентов системы. Она позволяет выделить внешние системы, контактирующие с системой, основные процессы и их взаимосвязь. Диаграммы прецедентов дают возможность выделить функциональную структуру системы, не вдаваясь в детали ее реализации. Кроме того, производится предварительное выделение объектов системы и их классификация. На основании построенной модели составляется план разработки системы.

Стадия системного проектирования включает в себя решения верхнего уровня относительно разработки системы в целом. Здесь производится разработка архитектуры системы с помощью диаграммы развертывания. Она позволяет выделить вычислительные ресурсы, устройства, использующиеся ими, и соединения между ними, спроектировать размещение отдельных процессов и исполняемых компонент на определенных устройствах, что особенно важно при проектировании сложных систем и интернет-приложений.

Для решения проблемы требуются специализированные вычислительные системы реального времени, смоделированные и, в дальнейшем, сгенерированные с помощью UML.

UML позволяет решать данную задачу путем разработки специализированного программного обеспечения и моделирования технических средств, которые могут быть реализованы на заказных БИС и СБИС [2]. С целью снижения сроков и стоимости создания перспективных образцов технических объектов (ТО), контроля и управления ими методы

проектирования должны удовлетворять требованиям высокого уровня формализации описания информационных процессов в контролируемых ТО и системах формирования и анализа воздействия на данные средства, адекватной оценки их состояния, а также возможности автоматизации их синтеза.

Из известных методов решения вышеназванных задач особое место занимают:

- теория синтеза автоматов, позволяющая решать проблему топологического проектирования БИС и СБИС, программируемых логических интегральных схем (ПЛИС), управляющих устройств [3];

- логико-комбинаторный метод синтеза структуры сложных объектов [4].

Метод синтеза цифровых автоматов [5] является эффективным методом проектирования логических схем, управляющих устройств. Однако для синтеза вычислительных структур, ориентированных на реализацию аналитических выражений небулевого характера и алгоритмов их реализации, этот метод является принципиально непригодным.

Использование логико-комбинаторного подхода к синтезу структур сложных систем, ориентирован на комбинацию известных технических решений с последующим выбором оптимальной структуры; таким образом, эта методология сводится к обыкновенному анализу и не обеспечивает нахождение нетрадиционных перспективных решений.

Известные системы автоматизированного проектирования позволяют решать задачи синтеза цифро-аналоговых, логических структур, трассировки, компоновки, выпуска технической документации (САПР P-CAD, AUTO-CAD и т.д.). В то же время отсутствуют средства автоматизации структурного и функционального проектирования, что связано, в первую очередь, с отсутствием до недавнего времени математических методов решения задачи.

Решение указанной проблемы становится возможным благодаря появлению теории синтеза вычислительных структур реального времени, разработанной А.С. Кобайло [6], основанной на назначении уровней временной иерархии вершинам графа вычислительного алгоритма реализации математической модели процесса и условия реализуемости пути графа в РМВ. Высокая степень формализации процесса проектирования сложных технических систем и возможность на основании этого его алгоритмизации позволит создать уникальную систему проектирования вычислительных структур РМВ. Обобщенный алгоритм, основанный на базовых положениях данной теории, представляет последовательность процедур реализации частных алгоритмов синтеза.

Исходными данными для алгоритма синтеза вычислительной структуры реального времени являются:

- 1) аналитическое выражение вычислительного процесса параметров модели;
- 2) условие требования реализации вычислительного процесса в реальном масштабе времени, в частности могут быть заданы временные характеристики обработки данных;
- 3) характеристики функциональных устройств, способных в совокупности реализовать все операции алгоритма исходной математической модели вычислительного процесса.

В результате реализации алгоритма, основанного на теории синтеза вычислительных систем реального времени, будут сформированы: граф вычислительной структуры и спецификация его вершин, транспонируемый в структурную схему проектируемой системы; вектор назначения, который может стать основой создания принципиальных схем; вектор временной развертки, являющийся основой синхронизации проектируемой системы и создания средств управления данной системы; такт вычислительной структуры, определяющей частоту следования тактовых импульсов.

Вектор временной развертки определяет программу взаимодействия функциональных устройств, составляющих вычислительную систему, на одном цикле ее функционирования [6].

Создание пакета прикладных программ синтеза систем реального времени на базе UML и сопряжение ее с лучшими мировыми образцами САПР предоставит возможность автоматизации всех стадий проектирования ВС и других технических систем, начиная с эскизного проекта и заканчивая выпуском технической документации, а также оптимального планирования технологических процессов.

Литература

1. Буч, Рамбо, Якобсон. UML.
2. Bruce Powell Douglass. Real-Time UML.
3. Скляр В.А. Синтез автоматов на матричных БИС. – Минск: Наука и техника. – 1984.
4. Кобайло А.С. Основы теории синтеза вычислительных структур реального времени. – Мн.: БГУИР, 2001.
5. Анкудинов Г.И. Синтез структуры сложных объектов: Логико-комбинаторный подход. – Л.: Изд-во Ленинград. Ун-та, 1986.
6. Кобайло А.С. Единичные функции для математических моделей вычислительных процессов: Деп. в Бел ИСА. - Мн.: РСНР, 1997.

АВТОМАТИЗАЦИЯ НА ОАО "САВУШКИН ПРОДУКТ"

Иванюк Д. С.

Брестский государственный технический университет, г. Брест

В конце 90-х было осознано, что дальнейшее развитие предприятия (в отношении производства) невозможно без автоматизации, поэтому встал вопрос - что дальше? Либо сторонние предложения, либо свои. Было принято решение разработать свою SCADA-систему. SCADA – Supervisory for Control And Data Acquisition - диспетчерское управление и сбор данных; SCADA-система – название класса ПО для создания систем автоматизации промышленного производства (АСУ ТП). В результате работы трех человек в течение 4-х лет была разработана программа Monitor и EasyServer. Один из первых проектов – контроль температурных режимов, в дальнейшем автоматизация приёмки молока и т.п. Сейчас – более 20-ти проектов, охватывающих различные области производства (начиная от учёта воды и электричества, различные цеха и участки), проекты в филиалах. Система имеет клиент-серверную структуру: на выделенном компьютере (сервере) выполняется серверная часть проекта, к нему подключаются клиенты (рабочие станции).

Общая схема системы управления приведена на рисунке 1.

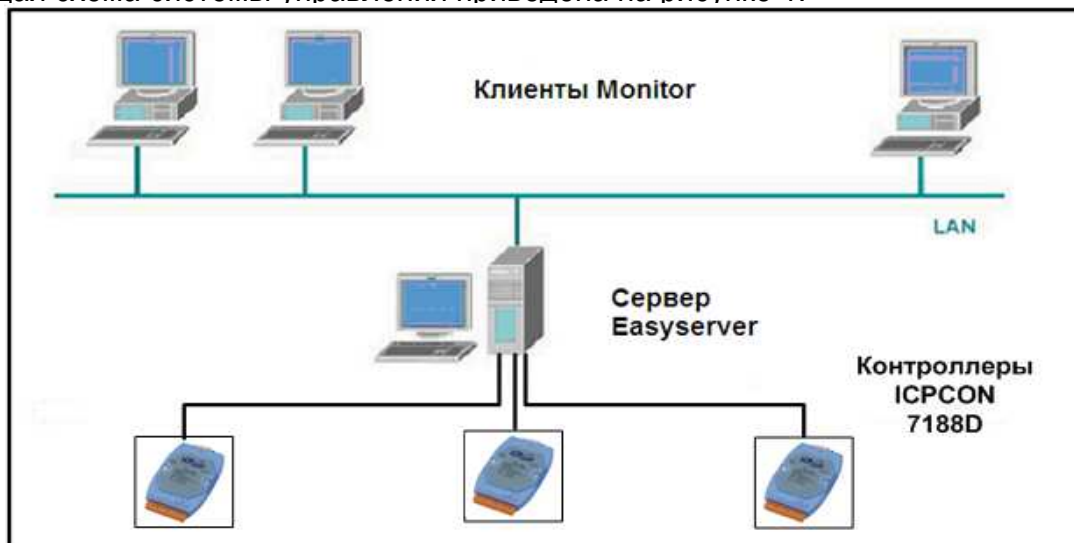


Рисунок 1