

КОНЦЕПЦИЯ ПРИМЕНЕНИЯ ГИБРИДНОГО МОДЕЛИРОВАНИЯ ПРИ РАЗРАБОТКЕ ПРОГРАММНЫХ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ

Погребной А. В.

Томский политехнический университет, Институт кибернетики,
avpogrebnoy@gmail.com

С развитием информационных технологий, создание и внедрение систем, функционирующих в режиме реального времени, приобрело массовый характер. Такие системы находят применение в самых разных областях. Назовём некоторые из них:

- управление техническими объектами и технологическими процессами (энергоблоки, реакторы, прокатные станы, конвейеры, производственные процессы);
- функции мониторинга, диспетчерского управления и оповещения (сеть гидрометеостанций, охрана лесов, сенсорные сети, экологический контроль);
- навигация и управление подвижными объектами (транспортные средства, летательные аппараты, необитаемые подводные аппараты);
- сервисное обслуживание (платежные системы, оказание услуг, социальные сети).

Системы для приведенных примеров объединяет то, что динамика их функционирования обусловлена тесным взаимодействием с внешней средой. Информация, поступающая в систему из внешней среды, должна обрабатываться и в виде результата возвращаться во внешнюю среду за ограниченное время. При этом масштаб времени в системе и внешней среде совпадает и соответствует реальному (календарному) времени. Такие системы выделены в отдельный класс систем реального времени [1].

В зависимости от последствий, которые наступают при нарушении ограничений реального времени, системы делят на жесткие и мягкие. В жестких системах, нарушения расцениваются как неисправность системы, в мягких – как снижение качества. Принято считать, что независимо от жесткого или мягкого варианта разрабатываемой системы, расчёт временных характеристик проще вести, рассматривая систему в жестком варианте, а по результатам расчёта оценивать соответствие техническому заданию.

Известные методы расчёта временных характеристик определяют локальные оценки и не могут дать полное представление о динамике функционирования системы в целом [1]. Методы имитационного моделирования здесь также неприемлемы из-за большой трудоёмкости разработки моделей. Для решения этой проблемы нужны экономичные и эффективные аналитические методы.

Важные результаты в этом направлении были получены в составе технологии структурного моделирования (SML-технологии) [2]. Они требуют развития, учитывающего современные возможности организации взаимодействия

системы с внешней средой. Для достижения этой цели необходимо решить следующие задачи:

- построение структурной модели системы;
- формализация процесса смены состояний модели системы в дискретные моменты времени;
- вычисление траектории смены состояний модели для совокупности параметров системы;
- анализ траектории смены состояний с целью вычисления временных характеристик и их соответствия требованиям технического задания.

Проектирование программных систем реального времени является сложной проблемой и требует от разработчиков проекта высокой квалификации. Это обусловлено рядом особенностей, характерных для систем данного класса. В таких системах прикладные задачи, как правило, имеют функциональное и территориальное распределение, которое отражает распределённость внешней среды. Это неизбежно приводит к сетевому варианту построения системы, что усложняет организацию взаимодействия между задачами.

На эти взаимодействия накладываются условия взаимодействия системы с внешней средой. Динамика событий, происходящих во ней, непосредственно устанавливает условия запуска задач в системе и требования к своевременному отклику. Таким образом, алгоритм функционирования системы должен постоянно адаптироваться не только к изменениям, происходящим внутри системы, но и во внешней среде. Такие условия функционирования системы требуют от разработчика проекта уделять самое пристальное внимание анализу гарантированного выполнения прикладных задач при соблюдении ограничений реального времени.

Что касается традиционных требований к разработке программного обеспечения (ПО) систем, таких как гибкость, масштабируемость, отказоустойчивость, эффективность, то для рассматриваемого класса систем необходимость соблюдения этих требований возрастает.

Приступая к созданию системы, разработчик проекта не в состоянии изначально спроектировать вариант системы, который реализует поставленные цели и отвечает необходимым требованиям и качеству. Поэтому при разработке системы приходится применять эволюционный подход, основанный на итерационном процессе совершенствования модели системы. Согласно этому подходу на первом шаге разрабатывается модель базового представления состава компонентов системы и отношений между ними, на основе которой программируется исходная

исполняемая функциональная модель. Результаты исполнения модели проверяются на соответствие базовому представлению и принимаются решения по расширению модели и соответствующей программной реализации. В новых моделях более полно учитываются задачи прикладных функций и требования к системе, а также уточняются детали их реализации.

Аналогично выполняются все последующие итерации эволюционного процесса. Цель каждой из них – получение рабочей версии ПО системы, включающей функциональность всех предыдущих и текущей итерации. Результат финальной итерации должен содержать всю требуемую функциональность ПО системы.

Модель базового представления системы на каждой итерации также расширяется и уточняется. Основное назначение модели – представление структурной составляющей семантики текущего состояния проекта системы. Такая модель позволяет разработчику воспринимать проект в целом и принимать решения по его расширению и совершенствованию.

Применительно к разработке систем рассматриваемого класса, использование эволюционного подхода имеет свои особенности. Они обусловлены спецификой функционирования таких систем и требованиями к качеству их работы. Выделим наиболее существенные из этих специфик и требований:

- большая неопределенность относительно финального состояния проекта системы как по архитектуре, так и динамике функционирования;
- динамика взаимодействия объектов внешней среды и системы регламентируется лишь частично и сопровождается непредсказуемыми моментами наступления событий;
- функционирование системы происходит в виде выполнения совокупности параллельных взаимодействующих процессов, что приводит к необходимости решения сложных задач синхронизации;
- повышенные требования к надежности и отказоустойчивости системы, что вынуждает разрабатывать средства автоматической реконфигурации системы.

Каждая из этих специфик, при проектировании системы требует решения проблем структуризации и анализа динамики функционирования. Решение данных проблем выполняется разработчиком проекта вручную, пользуясь моделью базового представления системы. Модель не является конструктивной и в данном случае применяется лишь как средство, удобное для визуального восприятия структуры системы. На такой модели нет возможности устанавливать зависимости между характеристиками системы и параметрами её компонентов.

Необходимость замены неконструктивной модели на конструктивную очевидна. Нужна

модель, которая сохраняет удобства визуального восприятия структуры системы и даёт возможность аналитическими методами определять характеристики системы и выполнять анализ динамики её функционирования. Такая модель названа структурной и представляется двудольным графом на языке структурного моделирования (SML) [2]. При этом в визуальные и функциональные возможности языка вносятся необходимые расширения.

В докладе предложена новая концепция применения структурного моделирования при разработке программных систем, функционирующих в реальном времени. Согласно этой концепции, разработка проекта системы ведется на основе эволюционного развития двух моделей – структурной (СМ) и функциональной (ФМ). Разработка ФМ отличается от обычного процесса программирования функционала системы внесением меток модульной структуризации проекта и соответствует получению исполняемой версии ПО текущего варианта системы. СМ строится одновременно с разработкой ФМ в соответствии с метками по правилам языка SML и визуально представляется двудольным графом.

На любом этапе эволюции ФМ может быть исполнена, а СМ использована для решения проблем синхронизации и анализа динамики работы системы с целью внесения в неё структурных и функциональных изменений. При структурном моделировании, большинство решений по эволюции проекта являются структурными и принимаются с помощью СМ. Поэтому, для разработчика проекта, СМ выступает в роли «путеводителя» при поиске приемлемого варианта системы, а ФМ – в роли «эксперта», оценивающего решения на очередной итерации проекта. В этой гибридной схеме моделирования на СМ возложена визуальная и аналитическая составляющие, а на ФМ – имитационная.

В целом, применение гибридной схемы моделирования даёт возможность параллельно с разработкой ПО системы оценивать качество её работы, что позволяет оптимизировать процесс развития проекта и снизить трудоёмкость и риски при его реализации. В будущих исследованиях по гибридному моделированию важно установить необходимый уровень соответствия между СМ и ФМ, а также степень влияния результатов аналитического и имитационного моделирования на достижение целей проектирования.

Список литературы

1. Гома Х. UML. Проектирование систем реального времени, параллельных и распределенных приложений: М.: ДМК Пресс, 2011
2. Погребной Ан.В., Погребной Д.В., Погребной В.К. Уравнения динамики функционирования CPB, представленной на языке SML. – LAP LAMBERT Academic Publishing, 2012. – 120 с.