

РАЗРАБОТКА МЕТОДИКИ ПРИМЕНЕНИЯ ИНФОРМАЦИОННЫХ МОДЕЛЕЙ ПРИ ПРОЕКТИРОВАНИИ СИСТЕМ ПРОМЫШЛЕННОГО ИНТЕРНЕТА ВЕЩЕЙ

Цуприянчик А.А. (Университет ИТМО, Санкт-Петербург)

Научный руководитель – к.т.н., доцент Третьяков С.Д.

(Университет ИТМО, Санкт-Петербург)

В представленной работе описан подход, основанный на моделях, для облегчения разработки адаптивных систем Интернета вещей. Разрабатывается методика, основанная на определении модели проектирования, функциональности системы, адаптациях и преобразовании модели Интернета вещей в код для развертывания на аппаратной платформе. В работе для моделирования функциональности используется расширение SysML (Eclipse Papyrus), для адаптации – подход конечного автомата, для генерации кода – Papyrus software designer.

Введение. В последние годы был предложен ряд подходов, облегчающих разработку систем Интернета вещей. Однако эти подходы не позволяют адаптировать системы Интернета вещей во время выполнения, пока они работают. Кроме того, некоторые из существующих подходов фокусируются только на одном этапе разработки (например, анализ или развертывание). Таким образом, существует необходимость в подходе, который полностью поддерживает разработку адаптивных программных систем Интернета вещей. Таким образом, целью данной работы является разработка методики применения информационных моделей при проектировании систем промышленного интернета вещей.

Основная часть. Papyrus – среда разработки на основе моделей в качестве решения этих проблем. Две основные концепции модельно-ориентированного проектирования, а именно абстракция и автоматизация, вместе являются лучшим решением двух первых проблем. Абстракция облегчает спецификацию и проектирование сложных систем IoT посредством моделирования.

Автоматизация позволяет управлять разнородными технологиями посредством автоматического преобразования моделей и генерации кода для конкретных технологических платформ. В дополнение к этим двум принципам принят общий архитектурный подход для расширения возможностей IoT: эталонную модель архитектуры IoT-A, которая предназначена для экстраполяции общих черт и определения уровня абстракции, который является общим для всех существующих архитектур, связанных с IoT.

Среда разработки и контроля на основе моделей IoT расширяет среду разработки на основе моделей Papyrus для определения, проектирования, развертывания и мониторинга системы IoT.

Papyrus для IoT использует методологию на основе IoT-A, чтобы направлять разработчика системы IoT во время разработки системы. Методология является общей и независимой от какого-либо продукта.

Фазы процесса разработки:

- создание модели системы, которая определяет функциональность системы и ее адаптивное поведение (реакцию системы на ряд ожидаемых изменений контекста);
- генерация кода, который соответствует проектной модели.

На первом этапе инженер-программист моделирует систему, способную справиться с заранее известными изменениями. На втором этапе модель системы преобразуется в реальную программную систему, которая может быть выполнена на виртуальной платформе для проверки адаптивного поведения системы.

Этапы разработки:

- 1) Определение функциональности системы и адаптивного поведения.

Система должна состоять из набора функций, которые взаимодействуют друг с другом для удовлетворения требований конечного пользователя.

2) Определение триггера адаптации и состояния системы (активное/неактивное).

Для моделирования адаптивного поведения системы необходимо указать как триггеры адаптации, так и состояния системы. Для каждого компонента определяется состояние системы времени выполнения.

3) Создание машины состояний.

Конечный автомат – это модель вычислений, основанная на гипотетической машине состояний. В один момент времени только одно состояние может быть активным. Следовательно, для выполнения каких-либо действий машина должна менять свое состояние.

4) Генерация кода.

Генерация кода выполняется в два этапа, где высокоуровневая модель системы преобразуется в конкретную модель интернета вещей, которая позже преобразуется в код.

Генерация кода обычно привязана к среде (платформе), в которой будет выполняться система, т.е. Rapyrus software designer является продолжением Rapyrus. Он обеспечивает генерацию кода из UML-моделей для ряда языков программирования, таких как C, C++ и Java, и поддерживает интеграцию новых генераторов кода.

5) Проверка программной системы.

Инженер-программист генерирует код "C". Симулятор запускается с помощью конфигурационного файла.

Когда симулятор запускается, он перемещает систему из заданного состояния в исходную конфигурацию. В мониторинге выполнения сценариев (мониторинг выполняется на аппаратной платформе) адаптации отображаются текущее состояние системы, триггер адаптации системы и выполнение планов адаптации.

Выводы. В этой работе была предложена методика для разработки таких систем IoT. Во-первых, на основе системных требований создается модель проектирования, которая фиксирует функциональность системы и ее адаптацию. Для моделирования адаптации среды выполнения используется подход конечного автомата/машины состояний. Во-вторых, модель используется для генерации системных реализаций. Для этого высокоуровневая Модель проектирования преобразуется в модель конкретной платформы Интернета вещей, которая затем используется для генерации Java-кода. Наконец, сгенерированный код развертывается на аппаратной платформе системы, чтобы сделать ее полностью функционирующей.

Цуприянчик А.А. (автор)

Подпись

Третьяков С.Д. (научный руководитель)

Подпись