

**Г.Л. МУРАВЬЕВ, С.В. МУХОВ, В.И. ХВЕЩУК**  
УО БрГТУ (г. Брест, Беларусь)

## **ОБ ИСПОЛНИМОСТИ СПЕЦИФИКАЦИЙ ПРОЕКТОВ**

При проектировании специализированного ПО, в том числе, с повышенными требованиями к надежности, срокам, стоимости проектных решений, широко используют высокоуровневые, поведенческие описания [1–3] и процедуры автогенерации проектных решений на их основе. При этом требуются верифицируемые спецификации, что обеспечивается и их исполнимостью [4, 5]. Например, в САПР аппаратуры, САПР ПО, в системах моделирования и т.д. Исполнимость спецификаций предполагает формализованное построение языка спецификации и программной поддержки интерпретации описания или получения загрузочных кодов. Первый подход проще в реализации в ущерб эффективности, что не является критичным для ряда систем, например, систем обучения. Второй сложнее в реализации, что связано с организацией лексического, семантического разбора описаний и генерации исполнимого кода.

Здесь рассмотрено применение методов, связанных с модификацией указанных подходов путем отказа, как от прямой интерпретации спецификации проекта, так и классической генерации загрузочного кода [6].

Модификация интерпретационного подхода использует возможности существующих компиляторов и позволяет достичь относительной простоты построения системы. Метод основан на получении внутренних, промежуточных представлений исходных спецификаций с последующим построением исполнимого кода. Внутреннее представление может быть получено путем автоматической генерации в процессе преобразования, трансформации исходного описания по заранее известным правилам в адекватный в функциональном отношении текст на внутреннем языке.

В качестве средства внутреннего представления описания выбирается универсальный ЯВУ общего назначения с развитыми вычислительными средствами, библиотеками и эффективным транслятором. Это позволяет далее максимально использовать возможности стандартных сред программирования и компиляторов. В качестве формата внутреннего представления используется текст, сформированный, например, на базе теоремы о структурированности текстов программ и метода введения псевдопеременной. Это позволяет представить любой исходный текст в виде управляемой последовательности тактов выполнения. Правила получения подобных описаний достаточно просты, а сами описания удобны для организации выполнения, включая пошаговое, реализации отладочных операций. Интерпретируемое описание в соответствии с указанной теоремой упрощенно может быть представлено в терминах управляющей структуры типа “выбор” как набор шагов, где команды каждого варианта выбора – шага

представляют группу операторов, выполняемых в одном такте, а текущее состояние исполнения и соответственно выбор группы исполняемых операторов определяется номером шага (такта) интерпретации.

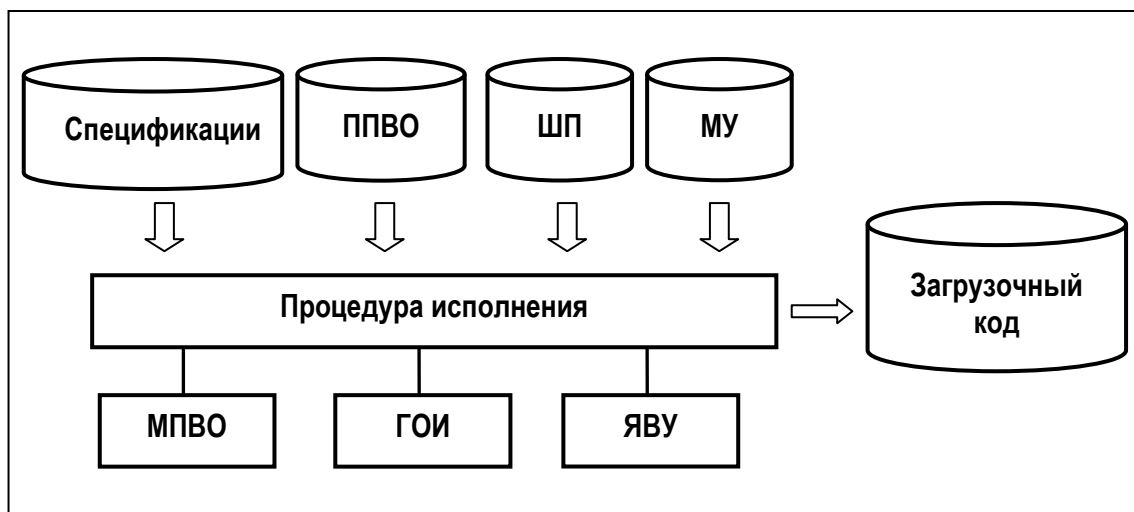


Рисунок 1. – Структура системы 1

Промежуточные описания в сборке с управляющим модулем, настроенным на соответствующее описание и реализующим потактовое выполнение, являются готовым продуктом для ЯВУ-трансляции и получения исполнимого кода. Для поддержки исполнимости нужны (рисунок 1):

- правила получения внутренних описаний (ППВО);
- модуль построения внутренних описаний, конвертор (МПВО);
- типовой транслятор ЯВУ;
- модуль управления потактовым исполнением (МУ),
- шаблон процедуры (ШП) и правила его настройки;
- генератор отладочной информации для поддержки процедур тестирования и отладки (ГОИ).

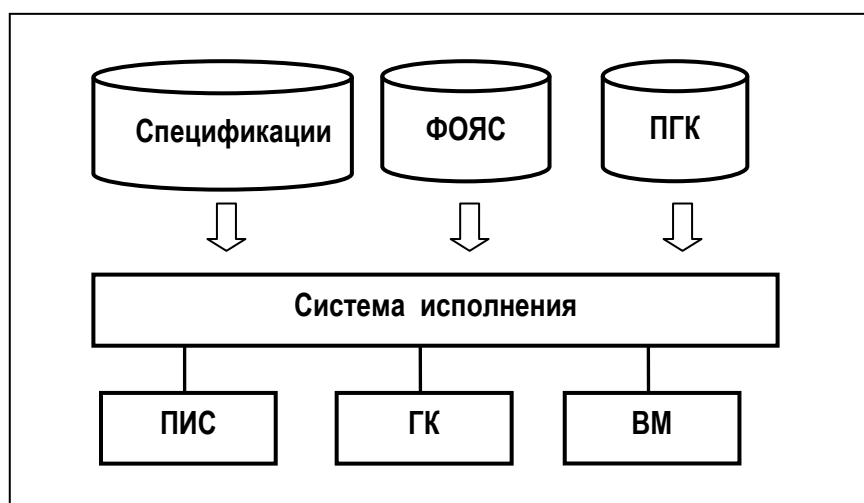


Рисунок 2 – Структура системы 2

Модификация компиляционного подхода базируется на принципах организации виртуальной машины (ВМ) и предполагает автоматическую генерацию промежуточного кода, интерпретируемого ВМ. Это упрощает реализацию системы, расширяет возможности диагностики и локализации ошибок и позволяет создавать полноценные и эффективные загрузочные коды. Промежуточное внутреннее представление спецификаций может быть получено путем проведения традиционных этапов их лексического, синтаксического и семантического анализа. В роли внутреннего представления результатов анализа используются структуры данных ЯВУ общего назначения, удобные для дальнейшей автоматической генерации функционально-адекватных описаний на внутреннем языке. В качестве формата внутреннего представления выбирается текст в терминах команд ВМ, что позволяет представить любой исходный текст в виде исполнимых команд. Здесь для поддержки исполнимости нужны (рисунок 2):

- формальное описание языка спецификаций (ФОЯС);
- правила генерации кодов ВМ (ПГК);
- препроцессор исходных спецификаций (ПИС);
- генератор кодов виртуальной машины (ГК);
- ВМ как “ядро” системы исполнения.

Примеры использования исполнимых спецификаций на базе модификации классических подходов приведены в [4; 5; 7; 8].

## ЛИТЕРАТУРА

1. Маклаков, С.В. Создание информационных систем с AllFusion Modeling Suite / С.В. Маклаков. - М: ДИАЛОГ-МИФИ, 2005. - 432 с.
2. Лисков, Б. Использование абстракций и спецификаций при разработке программ / Б. Лисков, Дж. Гатэг. - М.: Мир, 1989. – 424 с.
3. Мяцяшек, Л.А. Анализ требований и проектирование систем. Разработка информационных систем с использованием UML / Л.А. Мяцяшек: пер. с англ. – М: Издательский дом «Вильямс», 2002. – 432 с.
4. Муравьев, Г.Л. К разработке специализированного программного обеспечения / Г.Л. Муравьев, С.В. Мухов, В.И. Хвещук // Инновационные технологии обучения физико-математическим дисциплинам: материалы 6-й Междунар. науч.-практ. конф., Мозырь, 2014. – С. 205–206.
5. Муравьев, Г.Л. Построение моделей по описаниям, согласованным с процессным способом моделирования / Г.Л. Муравьев, В.И. Хвещук // Современные информационные компьютерные технологии mcIT-2008: сб. научн. ст., Гродно: ГрГУ, 2008. – Ч. 2. – С. 235–238.
6. Пратт, Т. Языки программирования: разработка и реализация / Т. Пратт, М. Зелковиц. – СПб.: Питер, 2002.

7. Муравьев, Г.Л. Автоматизация имитационного моделирования поведенческих описаний процессов / Г.Л. Муравьев, С.В. Мухов, В.Н. Шуть // Вестник БГТУ. – 2010. – № 5 (65). – С. 32–35.

8. Муравьев, Г.Л. Автоматизация обучения алгоритмизации и программированию / Г.Л. Муравьев // Вести ИСЗ. – 2004. – № 3. – С. 24–29.