

АДА-ТЕХНОЛОГИИ В 21 ВЕКЕ

НЕ МИФ, А РЕАЛЬНОСТЬ!

К сожалению, в российской программной индустрии сложился порочный круг: Ада-технологии не используются в промышленных проектах, поскольку они практически неизвестны как разработчикам, так и руководителям, принимающим ключевые технические решения. В результате отсутствует заказ вузам на специалистов, владеющих Адой. Как следствие, в индустрию приходят новые люди, ничего не знающие об Аде. В то же время Ада [1] является существенной компонентой мировой программной индустрии, использующейся при разработке больших долгоживущих встроенных систем реального времени. Ситуацию, когда Ада не используется в российской программной индустрии и отсутствует в ИТ-образовании просто потому, что она практически неизвестна отечественным специалистам, нельзя признать нормальной.

АДА-ТЕХНОЛОГИИ: МИФЫ И РЕАЛЬНОСТЬ

В лучшем случае российский ИТ-специалист просто ничего не знает об Аде. К сожалению, часто можно столкнуться с устаревшими и в корне неверными представлениями о языке, которые являются едва ли не большим препятствием к его использованию, чем полное незнание. Вот перечень основных мифов, связанных с Адой.

Миф: Ада – мертвый язык, некогда популярный, но сейчас нигде не используемый. **Реальность:** Ада всего лишь отсутствует на рынке "коробочных" продуктов. Язык активно применяется в области больших встроенных систем. Недавно была утверждена очередная ревизия стандарта Ады, а стандарты мертвых языков не пересматриваются.

Миф: Ада – язык, предназначенный исключительно для военных встроенных систем. **Реальность:** нет никаких формальных или технических причин, препятствующих применению Ады в других областях, число гражданских применений языка давно уже превосходит число военных применений, Ада – не менее, а даже более универсальный язык, чем

С.Рыбин, к.ф.-м.н.,
консультант компании AdaCore,
НИВЦ МГУ, Москва,
rybin@adacore.com

C/C++ или Java (например, среда разработки GNAT реализована на Аде).

Миф: Ада – слишком большой и тяжелый язык для использования в небольшом проекте. **Реальность:** достаточно сопоставить стандарты Ады и C++, чтобы убедиться, что определение Ады компактнее и проще в ситуации, когда по выразительным средствам Ада заметно превосходит C++.

Миф: Ада-технологии неэффективны. **Реальность:** современные промышленные реализации промышленных языков сопоставимы по эффективности, что легко проверяется на практике.

Миф: Ада-технологии слишком дороги. **Реальность:** здесь как раз тот самый случай, когда скупой платит дважды. Кроме того, существуют полноценные бесплатные версии Ада-технологий для использования в образовании и для разработки GPL-программ.

Миф: в России нет специалистов, знающих Аду, подготовка таких специалистов слишком дорога. **Реальность:** есть энтузиасты и коллективы, способные стать "точками кристаллизации" для Ада-проектов [2, 3]; обучение Аде немногим сложнее обучения Паскалю и существенно проще обучения C++; толковый программист, знающий Паскаль, способен освоить Аду в объеме, необходимом для начала практической работы, за неделю.

ПРЕИМУЩЕСТВА АДЫ ДЛЯ ИНДУСТРИИ

Ада – единственный язык, специально созданный для обеспечения надежности программного кода при разработке и сопровождении больших долгоживущих систем. Двадцатипятилетний опыт использования языка подтверждает, что как среднее число ошибок, так и время, проводимое разработчиками в отладчике при использовании Ады, существенно меньше, чем для других языков. Вот далеко не полный перечень языковых средств, особенностей



и механизмов, способствующих повышению надежности кода:

- строгая типизация при отсутствии неявных преобразований типов, часто оказывающихся абсолютно "неожиданными" для разработчиков;
- механизм подтипов и исключений;
- развитая модульность с полным статическим контролем межмодульных связей;
- удобные средства создания защищенных абстракций;
- ясный синтаксис с разумным уровнем избыточности, приводящий к тому, что грамотно написанный код на Аде читается как текст на формализованном английском языке.

Ада не просто не менее универсален, чем другие универсальные языки – она предоставляет уникальный набор возможностей для программирования асинхронных процессов. Средства работы с асинхронными процессами являются высокоуровневыми операторами языка, они отражают естественную логику поведения объектов данных и процессов в реальном асинхронном мире. В этой области у Ады сегодня просто нет достойных конкурентов.

Ада изначально возникла не как описание языка, а как стандарт языка, причем к моменту публикации стандарта уже были готовы средства контроля соответствия реализаций стандарту. В результате в Ада-мире отсутствуют какие-либо

диалекты языка, и перенос программной системы с одной реализации на другую требует на порядок меньше усилий, чем для других языков.

Ада изначально разрабатывалась с расчетом на разработку встроенных систем методом кросс-компиляции. Так, большинство механизмов Ады, связанных с надежностью, обеспечиваются проверками периода компиляции, не утяжеляя код, которому предстоит работать на целевой архитектуре.

Ада является кросс-платформенной технологией, промышленные реализации языка созданы и поддерживаются для всех промышленных платформ.

Ада предоставляет развитые средства интерфейса с модулями, написанными на других языках, существенно облегчая разработку многоязыковых систем.

Ада позволяет существенно сократить затраты на разработку средств анализа кода программ за счет интерфейса ASIS (Ada Semantic Interface Specification) [4], предоставляющего высокоуровневые средства доступа к синтаксису и семантике Ада-программ. Это может оказаться важным для тех областей, в которых часто требуется сертификация программного кода – основанные на ASIS инструменты могут автоматически выполнять существенную часть работы, связанной с сертификацией.

Этот перечень преимуществ Ады для индустрии далеко не исчерпывающий.

ИСПОЛЬЗОВАНИЕ В ОБЛАСТИ ВСТРОЕННЫХ СИСТЕМ

Индустриальные применения Ады в значительной степени находятся в области встроенных систем, среди них, в свою очередь, важное место занимают аэрокосмические приложения. Возможно, немногие знают, что при подготовке самолета Ил-86 к международной сертификации часть бортового программного обеспечения была переписана на Аде. Ада-технологии использовались специалистами Таганрогского авиационного научно-технического комплекса им. Г.М.Бериева при разработке амфибии Бе-200.

На сайте компании AdaCore (разработчика одной из основных индустриальных Ада-технологий – системы программирования GNAT) [5] приведен неполный список клиентов компании. Он включает такие имена, как Raytheon, Boeing, BAE Systems, EADS, Eurocontrol, Indra, Lockheed Martin, MBDA, Thales.

Там же названы некоторые проекты, реализуемые на Аде с помощью системы программирования GNAT, среди них:

- бортовая система управления транспортного самолета C-130J;
- система позиционирования для Airbus A350 XWB;
- система управления перископом для английских подводных лодок класса Astute;
- бортовое программное обеспечение для Boeing 787.

СИСТЕМА ПРОГРАММИРОВАНИЯ GNAT

Система программирования GNAT [5] – это одна из основных Ада-технологий, используемых сегодня в западной индустрии. Имеется положительный опыт применения GNAT по крайней мере в одном российском проекте в области разработки систем организации воздушного движения. Среди достоинств GNAT-технологии:

- система реализована на всех основных индустриальных платформах, причем ее пользовательский интерфейс одинаков на всех платформах;
- система позволяет создавать код для встроенных архитектур (ELinOS, Nucleus OS, PowerPC LynxOS, VxWorks, и др.), а также поддерживает технологию Bare Board;
- система поставляется с полностью открытым кодом;
- существует бесплатная версия системы для использования в учебных заведениях.

ПОДДЕРЖКА СТАНДАРТА DO-178B

Одним из основных направлений развития системы программирования GNAT в течение нескольких последних лет является поддержка требований стандарта Open DO-178B в области сертификации программных продуктов. В настоящее время GNAT-технология предоставляет ряд инструментов и решений, позволяющих существенно облегчить сертификацию кода.

Различные версии библиотеки периода исполнения (Run-Time Library – RTL): при сертификации программного продукта возникает проблема сертификации используемым этим продуктом библиотек, в число которых, как

правило, входит стандартная библиотека компилятора. Для решения этой проблемы, помимо версии RTL, определяемой стандартом Ады, GNAT предлагает несколько уровней RTL, отвечающих разным требованиям к сертификации кода. Так, минимальная версия библиотеки периода исполнения ZFP (Zero Foot Print) предоставляет минимальный набор возможностей, позволяющих создавать встроенные приложения, при этом затраты на сертификацию самой ZFP RTL минимальны. Версия RTL, удовлетворяющая профайлу Ravenscar, позволяет создавать приложения с асинхронными процессами.

Traceability Analysis Package: методика и набор инструментов, позволяющих определять для конкретной платформы набор языковых конструкций и параметров компиляции, для которых выполнение критериев структурного тестирования для исходного кода гарантирует выполнение тех же критериев для объектного кода.

Couverture: методика и набор инструментов, позволяющие в процессе тестирования автоматически проверять выполнение критериев структурного тестирования без модификации тестируемого кода.

Контроль стиля кодирования: набор инструментов, входящих в состав системы программирования GNAT, содержит инструмент gnatcheck, который проверяет выполнение для анализируемого кода различных правил, выходящих за требования стандарта языка и относящихся к стилям и стандартам кодирования. Набор правил постоянно расширяется, инструмент позволяет определять различные стили кодирования, комбинируя правила и их параметры.

"КОРРЕКТНОСТЬ ПО ПОСТРОЕНИЮ"

На базе Ады разработана и реализована одна из немногих практически успешных систем доказательного программирования – SPARK [6]. SPARK – это язык программирования, включающий подмножество Ады как средство описания действий. Также в него входят логические аннотации, описывающие свойства компонентов программы, и условия, которые должны проверяться в ходе ее выполнения. В средства реализации этого языка заложена возможность проверки истинности логических аннотаций, и успешная "компиляция" SPARK-кода есть формальное доказательство его корректности (под корректностью понимается истинность условий, сформулированных относительно заключительного состояния программы (постусловий), если выполнены условия, налагаемые на начальные данные (предусловия)). Исполняемый код можно получить, откомпилировав SPARK-код при помощи обычного Ада-компилятора, поскольку все аннотации представлены в виде Ада-комментариев специального вида. Существует положительный опыт применения технологии SPARK для реализации встроенных систем с критическими требованиями к надежности. Система программирования GNAT интегрирована с технологией SPARK.



АДА В ОБРАЗОВАНИИ

В современном российском ИТ-образовании Ада представлена в лучшем случае фрагментарно. В то же время Ада обладает рядом свойств, которые делают ее едва ли не лучшим кандидатом на роль базового языка в вузах, осуществляющих подготовку ИТ-специалистов. Ада (вместе с языками семейства Оберон) является достойной и естественной заменой безнадёжно устаревшему Паскалю. Вот далеко не полный перечень достоинств Ады с точки зрения образования:

- Ада обладает ясным и легко читаемым синтаксисом, что позволяет не отвлекаться на связанные с языком проблемы в ситуации, когда язык в рамках учебного курса используется для примеров и иллюстраций.
- Универсальность Ады позволяет использовать ее в большинстве современных курсов по ИТ, включая курсы по системам реального времени, асинхронному программированию, распределенным системам, технологии программирования и пр.
- Будучи специально спроектирована с целью обеспечения надежности кода, Ада представляет собой не только и не столько набор конструкций для написания программ, сколько целостный философский взгляд на разработку программного обеспечения.

- Существуют официально доступные бесплатные версии Ада-технологий, предназначенные для использования в учебных заведениях (Программа GNAT Academic Program, реализуемая компанией AdaCore).

Ада-технологии являются существенной компонентой мировой программной индустрии. Они используются в областях, в значительной степени определяющих научно-технический и экономический потенциал общества. И поэтому российские специалисты по ИТ должны иметь достаточно информации о современном состоянии Ада-технологий, чтобы принимать осознанные решения относительно их использования в тех или иных проектах.

ЛИТЕРАТУРА

1. Ada Reference Manual, ISO/IEC 8652:2007(E) Ed. 3.
2. ru.ada-community.org.
3. www.ada-ru.org.
4. Ada Semantic Interface Specification (ASIS), ISO/IEC 15291:1999(E).
5. www.adacore.com.
6. **John Barnes**. High Integrity Software: The SPARK Approach to Safety and Security. Addison-Wesley, 2003.