

АДА-ТЕХНОЛОГИИ В XXI ВЕКЕ – НЕ МИФ, А РЕАЛЬНОСТЬ!

Сергей Рыбин

К сожалению, в России Ада-технологии практически неизвестны ни разработчикам, ни руководителям, принимающим ключевые технические решения. В то же время Ада является существенной компонентой мировой программной индустрии, которая используется при разработке больших долгоживущих встроенных систем реального времени

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 для ВУЗов со стороны рынка отсутствует заказ на подготовку специалистов, владеющих Адой. Это приводит к тому, что в индустрию приходят новые люди, ничего не знающие об Аде. В то же время Ада является существенной компонентой мировой программной индустрии, которая используется при разработке больших долгоживущих встроенных систем реального времени. Ситуацию, когда Ада не применяется в российской программной индустрии, отсутствует в ИТ-образовании, и 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, которые являются едва ли не большим препятствием к использованию языка, чем полное его незнание.

Перечень основных мифов, связанных с Адой

Миф	Реальность
Ада – мертвый язык, некогда популярный, но сейчас нигде не используемый.	Ада всего лишь отсутствует на рынке «коробочных» продуктов. Язык активно применяется в области больших встроенных систем. Недавно была утверждена очередная ревизия стандарта Ады, а в настоящее время ведется активная работа над следующей ревизией, тогда как стандарты "мертвых" языков не пересматриваются.
Ада – язык, предназначенный исключительно для военных встроенных систем.	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 C/C++ или Java (например, среда разработки GNAT реализована на Аде).
Ада – слишком большой и тяжелый язык для использования в небольшом проекте.	Достаточно сопоставить стандарты Ады и C++, чтобы убедиться, что определение Ады компактнее и проще в ситуации, когда по выразительным средствам Ада заметно превосходит C++.
Ада-технологии неэффективны.	Современные промышленные реализации промышленных языков сопоставимы по эффективности, что легко проверяется на практике.
Ада-технологии слишком дороги.	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.

В России нет специалистов, знающих Аду, подготовка таких специалистов слишком дорога.

Есть энтузиасты и коллективы, способные стать «точками кристаллизации» для Ада-проектов; обучение Аде немногим сложнее обучения Паскалю и существенно проще обучения С++; 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 долго живущих 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 является не менее универсальным языком, 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 на данный момент 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 создавалась с расчетом на разработку встроенных систем методом кросс-компиляции. Так, большинство Адовских механизмов, связанных с обеспечением надежности, реализуются проверками периода компиляции, не утяжеляя код, которому предстоит работать на целевой архитектуре.

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

Ада предоставляет развитые средства интерфейса с программными 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), 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 выполнять существенную часть работы, связанную с сертификацией, автоматически.

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 КБ Бериева при разработке амфибии Бе-200.

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

Кроме того, там же названы некоторые проекты, реализуемые на Аде с помощью системы программирования GNAT, среди которых бортовая система управления транспортного самолета С-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, разрабатываемая, поддерживаемая и распространяемая начиная с 1996 года компанией AdaCore, является в настоящее время одной из основных Ада-технологий, используемых в западной индустрии. Имеется положительный опыт использования GNAT по крайней мере в одном российском проекте в области разработки систем организации воздушного движения. GNAT-технологии имеют целый ряд достоинств.

Во-первых, система реализована на всех основных индустриальных платформах, причем ее пользовательский интерфейс одинаков на всех платформах;

Во-вторых, система позволяет создавать код для встроенных архитектур (ELinOS, Nucleus OS, PowerPC LynxOS, VxWorks, и др.), а также поддерживает технологию Bare Board.

В-третьих, система поставляется с полностью открытым кодом под лицензией GPL, причем в профессиональной версии системы отсутствует проблема GPL-инфицирования, что позволяет разрабатывать приложения с закрытым кодом;

В-четвертых, для платформ Windows и Linux свободно доступна полная (то есть не содержащая каких-либо ограничений по функциональности и включающая базовый инструментарий) версия GNAT-технологии, позволяющая разрабатывать программы под лицензией GPL, в число свободно доступных конфигураций входит также кросс-компилятор, позволяющий генерировать код для платформы LEGO Mindstorms.

В-пятых, AdaCore предлагает бесплатную программу поддержки использования свободно распространяемой версии в учебных заведениях.

Поддержка стандарта DO-178B

Требования к надежности и безопасности программного обеспечения (ПО) в основных областях применения Ада-технологий, как правило, включают обязательную сертификацию программных систем. Для аэрокосмических приложений в качестве основы для программ и процедур сертификации часто используется стандарт DO-178B. Этот стандарт содержит проце-

дуры и требования к сертификации, которые имеют много общего с применяемыми в других системах сертификации ПО, требования к надежности которого являются критичными.

Сертификация ПО на базе DO-178B (или аналогичных стандартов) предполагает демонстрацию и обоснование того, что сертифицируемая система обладает рядом заявленных свойств. Сертификация - это длительная, трудоемкая и, следовательно, дорогостоящая процедура, требующая высококвалифицированного персонала. Значительные усилия тратятся на таких этапах сертификации, как:

- тестирование: для тестирования, проводимого в рамках сертификации ПО, во-первых, требуется выбор конкретного критерия полноты (обычно это тот или иной критерий структурного тестирования), а во-вторых, требуется доказать, что проведенная процедура тестирования обеспечивает выполнение выбранного критерия для тестируемого ПО;
- проверка соответствия кода ПО определенному стандарту кодирования, как правило, стандарт кодирования призван обеспечить отсутствие в тексте ПО нежелательных по тем или иным причинам языковых конструкций и их сочетаний (например, стандарт кодирования может запрещать конструкции, снижающие ясность и структурированность кода, ведущие к неконтролируемому росту динамической памяти и т.п.);

Стандарт DO-178B разрешает на определенных условиях использование программных инструментов для выполнения тех или иных действий, связанных с сертификацией ПО. Автоматизация наиболее ресурсоемких шагов процесса сертификации позволяет на порядки сократить требуемые временные, человеческие и финансовые ресурсы.

В ответ на пожелания клиентов предоставить решения и инструменты, облегчающие и автоматизирующие действия по стандартизации ПО, AdaCore специально разработала в рамках GNAT-технологии следующие решения:

Конфигурируемая библиотека 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, так как она практически не содержит исполняемого кода. Версия RTL, называемая Certified Profile, является сертифицированным расширением ZFP RTL, позволяющим разрабатывать практически любые встроенные приложения, не использующие асинхронные процессы. Версия RTL, удовлетворяющая профайлу Ravenscar, позволяет создавать приложения с асинхронными процессами, для которых может быть обосновано, что поведение системы всегда остается предсказуемым и детерминированным.

Статический анализ максимальной глубины стека: инструмент gnatstack, входящий в состав системы программирования GNAT, позволяет на основе статического анализа текста программы для каждого вызова подпрограммы определять максимально возможную глубину стека, а также найти вызовы, которые могут привести к неограниченному росту стека.

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. Более того, для встроенного приложения анализ полноты тестирования полностью проводится на инструментальной машине (за счет эмуляции выполнения программы в целевой среде).

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.

Стандарт DO-178B, разрешая применение программных инструментов и технологий в процессе сертификации программного кода, требует, чтобы для самих этих инструментов и технологий были представлены материалы, демонстрирующие, что данный инструмент или технология работает корректно (*qualification materials* в терминах DO-178B). Для всех перечисленных выше компонент GNAT-технологии в случае их использования в процессе сертификации пользовательского ПО могут быть представлены материалы, демонстрирующие их корректное функционирование для конкретной платформы и целевой среды.

"Корректность по построению"

На базе Ады разработана и реализована одна из немногих практически успешных систем доказательного программирования – SPARK. SPARK – это язык программирования, включающий подмножество Ады как средство описания действий и логические аннотации, описывающие свойства компонент программы и условия, которые должны выполняться в ходе ее выполнения. Реализация этого языка проверяет истинность логических аннотаций, и успешная "компиляция" SPARK-кода есть формальное доказательство его корректности (под корректностью понимается истинность условий, сформулированных относительно заключительного состояния программы (пост-условий), 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.

Образовательный аспект

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 безнадежно устаревшему Паскалю. Перечень достоинств Ады с точки зрения образования достаточно обширен.

Во-первых, 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;

Во-вторых, 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.

В-третьих, 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.

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

Заключение

Ада-технологии являются существенной компонентой мировой программной индустрии, применяемой 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.