

# ВОЗМОЖНОСТИ ОБЕРОН-ТЕХНОЛОГИЙ ДЛЯ СЛОЖНЫХ ЗАДАЧ СИСТЕМНОГО ПРОГРАММИРОВАНИЯ

*Ермаков И.Е.*

*ООО «Метасистемы»; Технологический институт ОрёлГТУ*

*302030, г. Орёл, ул. Московская, д. 34*

*e-mail: ermakov@metasystems.ru*

## **I. Введение**

За последние несколько лет системы программирования семейства Oberon Н. Вирта получили достаточную известность в нашей стране. Центрами консолидации Оберон-сообщества являются проекты Информатика-21 [1] и OberonCore.ru. Специалистами из науки, образования и индустрии, использующими Обероны, была сформулирована единая система взглядов относительно преимуществ применения Оберонов на всех этапах обучения программированию, а также для решения сложных научных и инженерных задач [1-9].

Как показала практика, основной и наиболее перспективной универсальной Оберон-системой для использования в промышленном программировании и обучении, является среда BlackBox Component Builder [10], которая разработана швейцарской компанией Oberon Microsystems [20] и уже несколько лет доступна по open-source-лицензии. В России среда поддерживается и развивается, в частности, силами компании ООО "Метасистемы" [11]. Среда BlackBox основана на языке Компонентный Паскаль, который, в свою очередь, является промышленной модификацией Оберон-2 (КП разрабатывался как альтернатива языку Java). В этой статье мы не будем проводить чёткой границы между "вообще Оберонами" и BlackBox (в силу соображений практики).

В силу предубеждений разного рода, сложившихся в массовом ИТ с 90-х гг., системе Паскаль-семейства трудно завоевать широкую популярность. Однако всегда остаются ниши сложных, ответственных работ, где очень важно качество инструмента: надёжность, защита от ошибок, простота - без которой невозможны первые два качества; концептуальная продуманность и отсутствие случайных исторических дефектов. Заметим, что задачи обучения программированию тоже безусловно относимы к категории сложных и ответственных. Системы Паскаль-семейства (Модула-2, Ада[12], Обероны) традиционно имеют крепкие позиции в таких областях, как встроенные системы, критичные по надёжности, с требованиями реального времени, и т.п. (как пример - технологии разработки бортового ПО спутников в НПО им.

Решетнева, основанные на Модуле-2 [13, 14]) Трудно считать разумным уход в угоду моде с этих качественных инструментов на системы из массового ИТ (например, такие тенденции, связанные с модой на язык Java, есть в ведомствах США, применяющих Аду).

Отметим, что общеевропейский проект ONBASS [16], направленный на повышение надёжности бортовых систем гражданской авиации, развивается с опорой именно на Оберон-системы (ОС А2[17] и др.) - разработки Native System Group университета ETHZ, Цюрих [18].

Что касается системы BlackBox Oberon, то она в своём непосредственном виде не готова к использованию во встроенных системах; однако зарекомендовала себя при создании инструментальных систем, научных и инженерных пакетов для этой сферы [19, 20] (например, пакет для моделирования - с расчётами и визуализацией - фазированных решёток прецизионных радаров для истребителя Eurofighter; разработка компании BAE Systems объёмом более 1 млн. строк исходного текста).

Кроме того, массовым результатом опыта BlackBox стала одна из наиболее популярных Java-машин для мобильных систем — Jbed от компании Esmertec [22, 23], которая некогда отпочковалась от Oberon Microsystems. Заметим также, что известный мировой специалист по компонентному программированию Клеменс Шиперски, с конца 90-х работающий в Microsoft Research и принимавший участие в создании платформы .NET, был в начале 90-х главным архитектором среды BlackBox [21]. Желаясь познакомиться с внутренним устройством современных компонентных платформ (frameworks), подобных .NET, можно смело рекомендовать изучить среду BlackBox.

## **II. Особенности возможности Оберон-технологий**

Что же скрывается за минимумом средств этого семейства языков, что позволяет говорить об их особенном положении среди инструментов программирования? Кратко ответим на этот вопрос, опираясь на собственный богатый опыт использования Оберона (язык Component Pascal в среде BlackBox). Во-первых, это динамичность и интеграция языка и системы разработки-выполнения, аналогичная интерпретируемым языкам, но сочетающаяся с «не-игрушечностью», высоким качеством каждой детали и прозрачной компиляцией в машинный код. Возможности динамической загрузки модулей, рефлексии, метапрограммирования не имеют аналогов среди компилирующихся систем.

Во-вторых, Компонентный Паскаль является идеальной систе-

мой для построения компонентных, эволюционирующих систем. Сама система BlackBox — живая иллюстрация компонентных паттернов (которые описаны в известной книге E. Gamma & ... “Design Patterns” [23]), сопоставимая в этом плане, например, со Smalltalk-системами. Набор средств современного модульного языка поддерживает мощные архитектурные решения, в том числе, не имеющие аналогов в mainstream (речь о шине типизированных сообщений на базе EXTENSIBLE RECORD). Подробнее об обероновских архитектурах можно прочесть в [4, 5]. Важно, что в язык и включено всё необходимое, и не внесены какие-либо «подводные камни», характерные для Java и .NET. (В этих системах существуют на первый взгляд безобидные упущения, которые обнаруживаются только при попытке выразить в них то, что прямым образом выполнялось на Обероне. В частности, паттерн «шина сообщений» в Java вообще не может быть реализован эффективно, из-за отсутствия стековых полиморфных структур данных. В C#.NET при попытке эмулировать модульность были допущены некоторые серьёзные дефекты. И т.п.)

В-третьих, Компонентный Паскаль предоставляет выразительный базис для структурирования данных, в современном понимании. В этом ещё одно (после компонентных паттернов) преимущество языка, органично расширенного от структурно-модульного до ООП (без введения отдельных понятий класса и объекта). Проектные решения в сложном алгоритмоёмком ПО — это отнюдь не только объединение данных и поведения (вошедшее в моду с ООП), но и аккуратное их расслоение, при сохранении на всех уровнях полиморфности и виртуализации. Здесь Обероны с их расширяемыми записями и сопряжёнными средствами качественно превосходят Java и C#. При этом в полной мере проявляется преимущество автоматического управления памятью при работе со сложными алгоритмами и структурами данных (часто можно говорить о принципиальном влиянии сборщика мусора на алгоритмику). Можно даже говорить, как это ни удивительно, о наличии в Обероне существенных аспектов, характерных для ФП. Такое наблюдение основано на нашем конкретном опыте разработки на Обероне транслятора функционального языка Рефал-0 [24] (ранее его высказывал координатор Информатики-21 Ф.В. Ткачёв, работающий в области вычислительных методов в физике высоких энергий).

И последнее наблюдение: Оберон прекрасно выполняет роль мостика через пропасть между задачами и оборудованием (про которую писал в своих работах Э. Дейкстра), давая прекрасные абстракции как для одной, так и для другой стороны. Если мы будем рассматривать программирование как продолжение прикладной математики, её замы-

вание обратно на «физическую реализацию», то Оберон выглядит очень естественным интеллектуальным средством, гладким продолжением мысли. Кроме прозрачности, надёжности и простоты здесь важна ещё возможность отображать идеи напрямую в конструкции языка, без долгого этапа подбора и «игр» на этом уровне. Это свойство Оберона повлияло не в последнюю очередь при выборе, который мы делали между ним и Адой (которая в плане продуманности и строгости также является языком высокого качества).

### **III. Оберон в современном системном программировании.**

Если выше мы излагали факты и выводы из собственного богатого опыта работы с Оберон-технологиями, то в этом разделе мы позволим себе поделиться мыслями относительно возможных перспектив этого направления.

Почему Оберон, заслуживший признания в штучных проектах, столь медленно и трудно продвигается в массовое программирование? Представляется, что дело здесь в том, что этот язык оказывается одинаково необычным и для системных, и для прикладных программистов.

Для системного программиста, воспитанного в 90-е годы на Windows и Unix, где доминирует Си-программирование, оказывается чуждым подход надёжных системных языков Паскаль-семейства (Модула-2, 3; Обероны; Ада). Однако эти языки позволяют разрабатывать без потери эффективности сложнейшие программные комплексы, обеспечивая на порядок меньшую плотность ошибок (устраняя ошибки «глупого» характера, связанные с нарушениями типов, разрушениями памяти, переполнениями буфера — на которых основаны практически все атаки вредоносного ПО). По разным оценкам, плотность ошибок в системах при использовании качественного языка может быть снижена в 10-16 раз. Однако лёд уже тронулся — слишком много ответственных программных систем окружает нас, и рано или поздно общество должно было озаботиться вопросами их надёжности. В частности, Евросоюз рассматривает проект закона об ответственности производителей ПО за ошибки в создаваемых системах. Такие экспериментальные ОС, как Microsoft Singularity, базируются снизу доверху на высокоуровневым безопасным языкам. И т. п.

Прикладные программисты выдвигают к Оберону претензии иного характера. Язык представляется им слишком низкоуровневым, «некомфортным», по сравнению с насыщенными «синтаксическим сахаром» C#, Java, Python и др. Однако постепенно осознаётся проблема избыточной сложности и удалённости от машины, характерной для

этих инструментов. Очень трудно предсказать реальное поведение системы, если между ней и машиной находится огромный «чёрный ящик» с непредсказуемой логикой работы. Надежды на постоянный рост уровня абстракций не оправдываются (точнее, этот рост даёт такие побочные эффекты — в виде сложности, неэффективности, непредсказуемости, - которые перекрывают все достижения). Видимо, имеет смысл говорить о наличии некоторого оптимального уровня абстракций, пролегающего между машинами и задачами. И нам представляется, что этот уровень выделен и представлен Никлаусом Виртом в его Обероне.

Выдвинем смелую мысль о том, что Оберон-системы можно называть «современным Юниксом», именно в силу простоты и прозрачности в отображении на машину. Удивительно, что при этом удаётся достигнуть полной независимости от низкоуровневых особенностей, герметичности системы типов, безопасности и удобных средств абстракции.

В Обероне мы можем подключить системные средства в конкретном модуле и спуститься на уровень Си и ассемблера, сохранив все гарантии безопасности для системы в целом. Мы можем полагаться на сборщик мусора как гарант правильных связей и времени жизни объектов в системе; но можем организовать работу с типизированными полиморфными данными на базе нединамических EXTENSIBLE RECORDs, размещаемых любым подходящим образом, - и организовать обработку огромных потоков данных, вообще не нагружая сборщик мусора (ср. с Java, где для создания любой структуры данных требуется размещение в куче; или с C#, где нединамические struct не поддерживают наследование и полиморфизм).

Сегодняшний день требует от нас ответственного программирования. Но невозможно отвечать за систему, опирающуюся на избыточно сложные монолитные компоненты с непредсказуемым поведением. Возможно, что вместе с взрослением нашей отрасли и приближением её к стандартам зрелых инженерных дисциплин мы увидим всё более широкое распространение простых, модульно организованных, доступных для понимания систем. И Оберон-технологии служат хорошим примером того, как это может выглядеть.

## Ссылки

1. Информатика-21 — <http://www.inr.ac.ru/~info21>
2. Система образования как фактор национального суверенитета в сфере информационных технологий - <http://www.inr.ac.ru/~info21/texts/>

[2006-09-SFO/v2public.pdf](#)

3. И.Е. Ермаков. Проблема обучения программированию и пути её решения. - <http://metasystems.ru/edu.php?page=pub>
4. И.Е. Ермаков. Оберон-технологии: что это такое? - <http://metasystems.ru/science.php?page=pub>
5. И.Е. Ермаков. Некоторые идеи архитектуры Оберон-систем. - <http://metasystems.ru/science.php?page=pub>
6. И.Е. Ермаков. Опыт применения Оберона при интеграции исследований, образования и производства. // Сборник трудов VII Международной конференции памяти академика Ершова «Перспективы систем информатики» (ПСИ-09), секция «Информатика образования». - <http://metasystems.ru/download/edu/I21-a-056-Novosib-PSI-2009-eie.pdf>
7. И.Е. Ермаков. О грамотной алгоритмизации. // Сборник трудов ПСИ-09, секция «Информатика образования» - <http://metasystems.ru/download/edu/I21-a-057a-Novosib-PSI-2009-eierbv.pdf>
8. И.Е. Ермаков. Презентация «Опыт построения систематического базового курса программирования». - <http://metasystems.ru/download/edu/I21-a-060-Novosib-PSI-2009-eie-P.pdf>
9. Р.П. Богатырёв. Заметки о программировании. <http://rbogatyrev.livejournal.com/2007/05/28/>
10. Краткая информация о BlackBox Component Builder - <http://oberoncore.ru/wiki/blackbox>
11. ООО «Метасистемы» - <http://metasystems.ru>
12. Русскоязычный ресурс по языку Ada и Ada-технологиям — <http://ada-ru.org>
13. А.А. Колташев. Модуля-2 в российском космосе. - <http://www.inr.ac.ru/~info21/texts/aakmodula2.htm>
14. А.А. Колташев. Эффективная технология управления циклом жизни бортового программного обеспечения спутников связи и навигации. - <http://www.inr.ac.ru/~info21/pdf/AAKpaper2006.pdf>
15. Отечественный ПК «Кронос» и ОС Excelsior - <http://kronos.iis.nsk.su/>
16. Проект Евросоюза по повышению надёжности ПО в гражданской авиации ONBASS - <http://www.oberon-industry.ethz.ch/projects/onbass>
17. Сайт швейцарской ОС Bluebottle (A2), язык Active Oberon - <http://bluebottle.ethz.ch/>
18. Native System Group, Университет ETH, Цюрих. - <http://www.nativesystems.inf.ethz.ch/>, <http://www.oberon.ethz.ch/>

19. Русскоязычный список применений Оберон-технологий - <http://oberoncore.ru/wiki/применения>
20. О применениях BlackBox на сайте Oberon Microsystems - <http://oberon.ch/references.html>
21. [http://oberoncore.ru/wiki/Oberon\\_Microsystems](http://oberoncore.ru/wiki/Oberon_Microsystems)
22. Компания Esmertec и Java-машина Jbed для мобильных устройств, изначально разработанная на базе BlackBox – <http://esmertec.com>
23. Э. Гамма и др. Приёмы ОО-проектирования. - СПб.: 2001.
24. И.Е. Ермаков. Встраиваемый язык обработки текстов Рефал-0 и разработка его транслятора на Компонентном Паскале. - <http://metasystems.ru/science.php?page=pub>