Руслан Богатырев

Оберон и виртуальная Java-машина

Первый компилятор Паскаля использовал промежуточный P-код, облегчавший его перенос между платформами. Почему в Обероне нет ничего подобного P-коду, хотя коммерческое направление программирования сейчас, наоборот, эксплуатирует эту идею (Java, .NET и т.п.)?

Прежде всего важно отметить, что первый компилятор языка Паскаль создавался не на Паскале, а на Фортране. Написание компилятора в 1969 г. Вирт поручил одному своему студенту (Эдуарду Мармье, Eduard Marmier). В тот момент Мармье владел лишь Фортраном и писал компилятор на этом языке с последующей трансляцией его в Паскаль. Затем компилятор Паскаля должен был подвергнуться процессу раскрутки (переписан на самом Паскале). Как отмечает Вирт в своем докладе "Recollection about the Development of Pascal" (ACM SIGPLAN History of Programing Languages Conference, Cambridge, USA, April 20-23, 1993), выбор Фортрана был серьезной ошибкой. Он не мог адекватно выражать сложные структуры данных компилятора, что все больше запутывало программу. Вторая попытка создать компилятор началась с того, что он сразу формулировался на самом Паскале (в соответствии с описанием 1970 г.). Синтаксический анализ нового однопроходного компилятора осуществлялся методом рекурсивного спуска. Теперь в команду разработчиков вошли Урс Амман (Urs Ammann), Эдуард Мармье и Рудольф Шилд (Rudolf Schild). После того как компилятор был написан на еще не существующем языке, Шилд был отправлен к себе домой на две недели, где все это время он вручную транслировал программу во вспомогательный низкоуровневый язык, доступный на мэйнфрейме CDC-6000. В середине 1970 г. компилятор ETH Pascal был готов. Он был интересен не только тем, что стал одной из первых реализаций языков высокого уровня на самом себе, примерно на два года опередив компилятор Си. В ходе работ над ETH Pascal в 1973 г. была придумана абстрактная Pascal-машина (P-машина), исполняющая специальный P-код. Чтобы решить проблему переноса компилятора Паскаля на разные платформы, Вирт решил воспользоваться испытанными временем методами интерпретации. Из наиболее известных решений, предшествовавших P-коду, можно назвать реализацию языка Snobol-4 (P. Грисуолд, 1967), где в качестве кода абстрактной машины использовался язык SIL (System Implementation Language).

Оберон при своем проектировании не предусматривал никакой трансляции в промежуточный код. Создавались компиляторы с генерацией машинного кода (native code) для вполне конкретных процессоров. В рамках проекта Oberon сам проф. Вирт писал однопроходный транслятор для NS32xxx (National Semiconductor), положенного в основу Oberon-компьютера Ceres. Для того же Ceres делался кодогенератор под StrongARM. В период 1989-1994 гг. были сделаны кодогенераторы для CICS- и RISC-процессоров: i386, SPARC, Motorola 680x0, PA-RISC, Power, PowerPC и др., причем почти все они — для компилятора OP2 (Оберон, Оберон-2).

Технология генерации внутреннего промежуточного представления (кода) давно и хорошо известна разработчикам оптимизирующих трансляторов. Это был конек известного семейства TopSpeed-компиляторов (Assembler, C/C++, Pascal, Modula-2, Ada, Clarion) фирмы JPI (Jensen & Partners International, затем TopSpeed Consortium, затем Clarion Software, наконец SoftVelocity), вышедших из недров Borland International. Свое более чем приличное ноу-хау в этой области имеет и новосибирская компания Excelsior (инструментарий XDS).

Внешнее промежуточное представление (P-код Паскаля, M-код Modula-2, Warren Abstract Machine для Пролога, деревья Франца для императивных языков, байт-код Java, MSIL в .NET и т.п.) ориентировано на переносимость (мобильность) исполняемого кода и последующую оптимизацию с привязкой к целевой процессорной архитектуре. Все более популярной становится двухфазная схема: SL → IL, IL→ TL (Source Language, Intermediate Language, Target Language). При этом IL (промежуточный язык) может интепретироваться, частично компилироваться (схемы JIT, DAC), полностью компилироваться (схемы WAT, AOT). Так, например, Excelsior JET является весьма качественной оптимизирующей реализацией AOT-компиляции Java.

В 1994 г. аспирант Вирта, Микаэль Франц, разрабатывавший ранее кодогенератор для MC680x0, завершил кодогенератор в промежуточный код — OMI (Oberon Module Interchange), где вместо устаревшего и уязвимого подхода на основе байт-кода использовался подход на основе семантических деревьев. Впервые на русском языке информация об этом была опубликована в отечественном альманахе "Технология программирования" (1995, No.1). Идея Франца была проста — вместо традиционной схемы "компилятор — компоновщик — загручик"

получить схему "компилятор — кодогенерирующий загрузчик", разделив фазы анализа и синтеза, иными словами, совместив на фазе синтеза генерацию кода, компоновщик (редактор связей) и загрузчик.

Концепция "code-generation on-the-fly" (динамическая кодогенерация, кодогенерация на лету) была положена в основу одноименной диссертации М. Франца, которую он защищал в ETH в начале 1994 г. Его научными руководителями были Никлаус Вирт и Юрг Гуткнехт. Редкий случай — в Цюрихе в марте 1994 г. она была переиздана в виде книги.

В Sun не стали полностью копировать все из Oberon (идеи браузерной среды языка, аплетов и трансляции в мобильный код взяли, а вот путь реализации мобильного кода выбрали допотопный). В 1991 г. автор Java Джеймс Гослинг при реализации Oak (прототипа языка Java) взял старую идею P-кода (переносимый Паскаль-код, созданный группой Вирта), которую хорошо знал: в 1975 г. Гослинг вместе с Недом Китлицем и Бобом Сайдботемом участвовал в построении среды программирования Pyxis/Multics Pascal, способной по быстродействию кода и удобству интеграции на равных конкурировать в Multics с родным для этой ОС языком PL/I. А начинали они с поддержки компилятора ETH/Zurich Pascal, разработанного в Цюрихе группой профессора Вирта. В 1979 г. Гослинг реализовал PERQ — транслятор с P-кода в машинный код DEC VAX.

В 1994 г. Sun решили не рисковать включением новейшей хитроумной реализации мобильного кода в древовидном представлении (что предлагал в диссертации Франц), а сохранили готовый подход Гослинга. Для всей отрасли модель Sun на долгие годы стала эталоном.

Франц вместе с Томасом Кистлером перенесли идеи OMI в среду Netscape и Internet Explorer для демонстрации возможностей своей технологии, но маркетинговая машина Sun с помощью IBM (чьи интересы против Microsoft тогда совпадали) заработала на полную катушку. В ответ Франц создал для демонстрации своего механизма среду Juice. Juice — это перенос Mac'овских наработок Франца в среду браузера. Это компилятор языка Oberon с механизмом OMI и небольшой базовой Оберон-библиотекой. Все это выполнено в 1996 г. в виде plugin (npJuice.dll) для упомянутых браузеров и занимает около 400 Кбайт. Есть все исходные тексты (на C++ реализована прослойка для ОС и браузера, на Обероне — все остальное). Разумеется, есть полный доступ к WinAPI, модель безопасности кода можно спокойно сделать свою.

В 1997-1998 гг. Франц опубликовал две свои работы в российских изданиях: "Мире ПК" и еженедельнике ComputerWeek-Moscow ("Java: критическая оценка" и "Есть ли у Java альтернативы?"). Они доступны на сайте Oberon2005 (на русском и английском языках).

В настоящее время Микаэль Франц является одним из ведущих в мире специалистов по обеспечению надежности и безопасности мобильного Java-кода и в университете Калифорнии в Ирвайне (США) руководит проектами в этой области.

Готовится представление проекта Juice на Oberon2005, включая документацию, исходные тексты системы и Оберон-компилятора, а также инструментарий для создания аплетов в рамках ETH Oberon System 3.