

С. Б. Покровский
ВНУТРЕННИЙ ЯЗЫК В БЕТА

Проектные установки

Проект БЕТА был основан на предположении о конвергенции основных алгоритмических языков, как и методов трансляции и оптимизации. Подтверждением тому казалось значительное внешнее сходство тогдашних “хитов” — Алгола 68 и ПЛ/1, как и стоявшей несколько особняком Симулы-67.

Проект был задуман как сочетание научно-исследовательской и опытно-конструкторской работы, предпосылкой к этому должен был стать успешный опыт разработки Альфа-системы.

Входные языки

Первоначально основными входными языками считались Алгол 68 и ПЛ/1.

Сентиментальная привязанность к Алголу была обусловлена традицией, восходящей к Альфа-проекту и общей склонностью советской школы программирования к идейно чистым и элегантным построениям. Алгол 68 должен был стать стержнем всей конструкции, благодаря провозглашенным им принципам ортогонального, абстрактного (машинонезависимого) и алгоритмического (высокоуровневого) построения.

ПЛ/1 был данью прагматическим соображениям. Во многих отношениях он был прообразом Си — тоже своего рода гигантским языком ассемблера, долженствовавшим предоставить доступ ко всем возможностям базовой вычислительной системы (конкретно, ИБМ/360). С другой стороны, детище ИБМ, корпорации, которая занимала почти монопольное положение на рынке (хотя и в меньшей степени, чем теперь Майкрософт), ПЛ/1 имел и ряд монструозных черт, напоминающих нынешние Windows: эффектных, броских, кажущихся удобными — но по существу вредных (вроде определения атрибутов по умолчанию, которое, поощряя неряшливое написание программ, порой приводило к чудовищно нелепым результатам).

Следует также учитывать, что в течение ряда лет заказчиком работ по проекту БЕТА выступала организация, ответственная за реализацию ЕС ЭВМ (советского аналога серии ИБМ/360/370); на практике, однако, это не изменило второстепенного положения языка ПЛ/1 в проекте БЕТА.

С самого начала в списке входных языков фигурировала также Симула-67. Этот язык казался схожим со своими “большими братьями”, упомянутыми выше — как благодаря общему происхождению от Алгола 60, так и вследствие неотличения квазипараллелизма Симулы от средств управления многозадачностью языков Алгол 68 и ПЛ/1. На деле Симула, несмотря на свой консервативный синтаксис (она сохраняла обратную совместимость с Алголом 60), была языком совершенно другой природы, на два поколения впереди Алгола 68 и ПЛ/1 (языком следующего поколения стал Паскаль, но и в нем отсутствовали классы и префиксация Симулы, распространившиеся с языками объектно-ориентированного программирования).

Однако тогда Симула казалась интересным, но не очень важным языком.

Язык Паскаль появился на свет одновременно с началом работ над проектом БЕТА и вскоре был включен в список его входных языков как основной входной язык “пробной реализации” БЕТА-системы.

Не всё было честно в громадном успехе Паскаля, однако он бесспорно ознаменовал поворотный пункт от эры динозавров, вроде Алгола 68 и ПЛ/1, к эре мелких грызунов. До Паскаля каждый новый язык претендовал на расширение репертуара изобразительных средств, на всё большую вычислительную мощь — но уже Алгол 68 провозгласил принцип отбора, ортогонального построения. Паскаль выставляет своими приоритетами простоту для транслятора и ясность для программиста — требования взаимосвязанные и взаимодополняющие. На смену оптимизму и быстрым успехам 60-х пришли скучные пуританские добродетели 70-х гг. прошлого столетия.

В итоге от языкотворчества 60-х не осталось ничего. Парадоксально, но более живучими оказались языки, возникшие еще в 50-х: Фортран и Лисп — хотя и после многочисленных изменений, сходных с эволюцией естественных языков.

Внутренний язык

Центральным звеном проекта БЕТА был Внутренний язык, который должен был стать единым языком-посредником в БЕТА-системе, “наибольшим общим делителем” входных языков и “наименьшим общим кратным” выходных машин. Кроме этой своей роли промежуточного языка, позволяющего уменьшить число путей в схеме m -языковой n -машинной трансляции с $m \cdot n$ до $m+n$, внутренний язык должен был также явиться средой оптимизирующих преобразований, т.е. он еще должен был быть достаточно богат, чтобы на нем было возможно представить результаты оптимизации; например, экономии совпадающих подвыражений в операторе $a[i,j,k] := b[i,j,k] + c[i,j,k]$.

Разработка внутреннего языка шла в три этапа.

Вначале А. А. Берсу было поручено проработать “идеальный” внутренний язык, каким его желательно было бы видеть с точки зрения анализа программ, для выявления чистой сущности алгоритма обрабатываемой программы.

Параллельно М. И. Шварцман должен был разработать более надежный (более обыкновенный) запасной вариант. Однако первоначально “идеальный” вариант считался основным.

А. А. Берс построил интересную модель вычислений, управляемых потоком данных (информационными зависимостями), в значительной степени ориентируясь на задачу максимального раскрытия внутреннего параллелизма задачи (задача автоматического распараллеливания была одним из приоритетов проекта). Схема получилась интересная, и наверно она могла бы стать основой перспективного проекта, если бы она была ведущим элементом проекта, под который подбирался бы входной язык и конструировалась бы выходная машина. Однако в рамках БЕТА-проекта языки и машины были данностью, а внутренний язык не определяющим, но производным элементом, и было непонятно, как уложить, например, чрезвычайно жесткую, почти ассемблерную модель памяти ПЛ/1 в эту идеальную схему распределенной памяти.

Проект М. И. Шварцмана предполагал наличие изобразительных средств для таких конструкций низкого уровня, и в конце концов это определило смену приоритетов и переориентацию на более прагматический подход. Но инерция ли общего замысла проекта, или воспитание, или ориентация на научный результат (естественная ввиду диссертационных планов) — что-то резко изменило направленность работы М. И. Шварцмана, и с момента принятия его проекта за основу, он

быстро стал обрастать чертами идеально-научной разработки, хотя и не такой идейно чистой, как проект А. А. Берса.

Эта работа, однако, не была доведена до конца. М. И. Шварцман покинул проект, и дальнейшая ответственность за разработку внутреннего языка пала на С. Б. Покровского (автора этих строк).

Фактически работа была начата заново и на сей раз не “сверху вниз”, а “снизу вверх”. Было реализовано подмножество Фортрана, и оно дало некоторый каркас, на который понемногу наращивалась плоть сообразно с потребностями подключаемых входных или выходных языков или алгоритмов оптимизации. Язык был наконец зафиксирован в виде структур данных (в общем виде программа на внутреннем языке представляла собой граф, и линейно-текстовая форма для визуализации частей этого графа появилась лишь через некоторое время). Вместо идеальной модели вычислений, или универсального набора семантических сущностей для содержательной унификации, внутренний язык стал частично-интерпретированной схемой, что позволяло выполнять оптимизационные преобразования, но требовало конкретизации при подключении новых языков или машин. Ситуация довольно обычная в производственном программировании, но менее эффективная, чем было задумано.

Однако этот путь позволил реализовать часть задуманного, и маловероятно, что чистая полноинтерпретированная модель оказалась бы совместимой с заданным набором реальных входных и выходных языков.

Итоги

В итоге система БЕТА была реализована для языков Симула-67, Паскаль, Модула-2, Ада (подмножество) и выходных машин БЭСМ-6 и СМ-4. Был реализован скромный набор оптимизаций — несмотря на обширные замыслы, более скромный, чем в системе Альфа. В общем, сравнительно с Альфа-системой, проект БЕТА следует признать неудачным.

Отчасти тому были “субъективные” (т.е. личностные) причины; но также и сам замысел был слишком рискован и основан на неверных посылах.

Замысел был слишком рискован, так как одновременно предполагалось опробовать слишком много новшеств: многоязычие на входе, многомашинность на выходе; новые языки (например, при начале работ

существовали только неполные экспериментальные трансляторы с Алгола 68) и новые методы оптимизации для новых прикладных областей — и всё это в рамках научно-исследовательского института, полупроизводственная работа силами сотрудников, ориентированных на получение научных результатов.

Неверная установка состояла в экстраполяции успешного развития вычислительного дела конца 50-х — начала 60-х гг. Казалось, что после успешного освоения проблематики численных методов, для которой удалось создать удобные изобразительные средства и эффективные алгоритмы оптимизации, подобных же результатов можно будет добиться и в области символической обработки, системного программирования, искусственного интеллекта и т.д. Увы, эти области оказались более вязкими, а достижения в автоматизации программирования — гораздо более скромными, чем ожидалось в 60-е гг. Но этот просчет касается не только проекта БЕТА.

В целом, в подобной ситуации я бы рекомендовал яснее определять приоритеты и закладывать в проект меньше тонких мест. Другой вопрос, как при этом получить финансирование.