

Языки паттернов Александра

Наша работа напоминает работу Александра во многих отношениях. Обе основаны на изучении существующих систем и нахождении в них паттернов. И там, и там есть шаблоны для описания паттернов, хотя и совершенно различные. Работы основаны на естественном языке и примерах, а не на формальных языках. В обеих для каждого паттерна приводятся обоснования.

Но во многих отношениях наши работы различаются:

- люди строят здания много тысяч лет, так что существует множество классических примеров. Программные же системы мы начали создавать сравнительно недавно, и лишь немногие признаны классикой;
- Александр приводит порядок, в котором следует использовать его паттерны, мы – нет;
- в паттернах Александра упор сделан на проблемах, в то время как паттерны проектирования гораздо подробнее описывают решение;
- Александр утверждает, что его паттерны способны сгенерировать проект всего здания. Мы не считаем, что наши паттерны могут создать законченную программу.

Когда наш коллега (в определенном понимании этого слова) говорит, что можно спроектировать здание, просто применяя паттерны один за другим, то он преследует те же цели, что и методисты объектно-ориентированного проектирования, которые приводят пошаговые правила. Александр не отрицает творческого подхода; некоторые из его паттернов требуют понимания привычек и обычаев людей, которые будут жить в здании, а его вера в «поэзию» проектирования подразумевает, что уровень проектировщика отнюдь не должен ограничиваться владением языком.¹ Но из его описания того, как паттерны генерируют проект, следует, что язык способен сделать процесс проектирования детерминированным и повторяемым.

Точка зрения Александра помогла нам сместить акцент на компромиссы проектирования – различные «силы», которые формируют дизайн. Под влиянием этого человека мы упорно трудились над тем, чтобы понять применимость и последствия каждого из наших паттернов. Идеология работы Александра уберегла нас от волнений по поводу формального представления паттернов. Хотя такое представление, возможно, помогло бы автоматизировать паттерны, мы полагаем, что на данном этапе важнее исследовать пространство паттернов проектирования, а не формализовать его.

С точки зрения Александра, описанные в этой книге паттерны не составляют языка. Принимая во внимание многообразие программных систем, трудно представить себе, как предложить «полный» набор паттернов, из которого можно было бы вывести пошаговые инструкции по созданию приложения. Для некоторых классов приложений, скажем, систем генерации отчетов или ввода данных путем заполнения форм это возможно. Но наш каталог представляет собой лишь набор взаимосвязанных паттернов, мы не претендуем на создание языка паттернов.

¹ См. "The poetry of the language" [AIS+77].

Нам кажется маловероятным, что *когда-либо* будет создан полный язык паттернов для проектирования программ. Но, безусловно, можно создать каталог *более* полный, чем наш. В него можно было бы включить описание каркасов и способов их применения [Joh92], паттернов проектирования пользовательского интерфейса [BJ94], паттернов анализа [Coa92] и прочих аспектов разработки программ. Паттерны проектирования – это лишь часть куда более широкого языка паттернов в программном обеспечении.

Паттерны в программном обеспечении

Нашим первым коллективным опытом изучения архитектуры программного обеспечения было участие в семинаре OOPSLA '91, который проводил Брюс Андерсон (Bruce Anderson). Семинар был посвящен составлению справочника для архитекторов программных систем. Данное событие положило начало целой серии встреч. Последняя из них состоялась в августе 1994 г. на первой конференции по языкам паттернов программ. В результате сформировалось сообщество людей, заинтересованных в документировании опыта разработки программного обеспечения.

Разумеется, ту же цель видели перед собой и другие исследователи. Книга Дональда Кнута «Искусство программирования для ЭВМ» [Knu73] была одной из первых попыток систематизировать знания, накопленные при разработке программ, хотя акцент в ней был сделан на описании алгоритмов. Но даже эта задача оказалась слишком трудной, так что работа осталась незаконченной. Серия книг *Graphics Gems* [Gla90, Arv91, Kir92] – еще один каталог, посвященный проектированию, хотя и он в основном посвящен алгоритмам. Программа Domain Specific Software Architecture (Архитектура проблемно-ориентированного программного обеспечения), которую спонсирует Министерство обороны США [GM92], направлена на подбор информации архитектурного плана. Исследователи, занятые разработкой баз знаний, стремятся отразить накопленный опыт разработок. Есть и много других групп, задачи которых в той или иной мере сходны с нашими.

Большое влияние на нас также оказала книга Джеймса Коплиена *Advanced C++: Programming Styles and Idioms* [Cop92]. Описанные в ней паттерны в большей степени, чем наши, ориентированы на C++. Кроме того, в книге приводится много низкоуровневых паттернов. Коплиен всегда был активным членом сообщества проектировщиков, заинтересованных в паттернах. Сейчас он работает над паттернами, описывающими роли людей в организациях, занятых разработкой программ.

Одним из первых, кто стал популяризировать работы Кристофера Александра среди программистов, был Кент Бек (Kent Beck). В 1993 г. он начал вести колонку в журнале *The Smalltalk Report*, посвященную паттернам в языке Smalltalk. Некоторое время паттерны собирал Питер Кoad (Peter Coad). В основном в его работе [Coa92] представлены паттерны анализа.

6.4. Приглашение

Что можете сделать лично вы, если вас интересуют паттерны? Прежде всего применяйте их и ищите другие паттерны, которые лучше отражают ваш подход