

Б.К.Шаяхметова<sup>1</sup>, Р.А.Сыздыкова<sup>1</sup>, К.С.Шаукенова<sup>1</sup>, Ю.Н.Антипов<sup>2</sup>

<sup>1</sup>Карагандинский государственный университет им. Е.А.Букетова;

<sup>2</sup>Калининградский технический университет, Россия (E-mail: yunikan@yandex.ru)

## Вопросы проектирования программ блочно-иерархический подход, постановка задачи

Для создания программного продукта сложной системы необходимо сначала разбить ее на более мелкие части (процесс декомпозиции), а затем сделать постановку задачи. Подавляющее большинство сложных систем как в природе, так и в технике имеет иерархическую внутреннюю структуру. В статье освещены приемы, применяемые при создании программного обеспечения, рассмотрен блочно-иерархический подход, сформулированы основные положения постановки задачи, показан процесс декомпозиции сложного программного комплекса.

*Ключевые слова:* иерархия, блок, декомпозиция, проектирование, программный комплекс, структура, процесс, объект, принципы, «простое — сложное».

В настоящей работе приведены приемы, применяемые при создании качественного программного обеспечения. Она является продолжением идей, заложенных в статье [1], в которой исследовались вопросы преподавания технологичности программирования.

Для того чтобы создать программный продукт для сложной системы, необходимо сначала разбить ее на более мелкие части (процесс декомпозиции), а затем сделать постановку задачи.

Рассмотрим блочно-иерархический подход и сформулируем основные положения постановки задачи.

Известно, что подавляющее большинство сложных систем как в природе, так и в технике имеет иерархическую внутреннюю структуру. Это связано с тем, что обычно связи элементов сложных систем различны как по типу, так и по силе, и это позволяет рассматривать системы как некоторую совокупность взаимозависимых подсистем. Внутренние связи элементов таких подсистем сильнее, чем связи между подсистемами. В качестве примера можно привести либо компьютер, состоящий из процессора, памяти и внешних устройств, либо солнечную систему, которая включает солнце и планеты, вращающиеся вокруг него.

Разделение на подсистемы, как различие связей, дает возможность каждую подсистему разделить на подсистемы до самого нижнего «элементарного» уровня, причем выбор уровня, компоненты которого следует считать элементарными, остается за исследователем. На элементарном уровне система, как правило, состоит из немногих типов подсистем, по-разному скомбинированных и организованных. Иерархии такого типа получают название «целое–часть».

Поведение системы в целом оказывается сложнее поведения отдельных ее частей, причем из-за более сильных внутренних связей особенности системы в основном обусловлены отношениями между ее частями, а не частями как таковыми.

Приведем пример, в нашем случае относительно иерархии имеет место следующее: в природе существует еще один вид иерархии — иерархия «простое — сложное», или иерархия развития (усложнения) систем в процессе эволюции. В этой иерархии любая функционирующая система является результатом развития более простой системы. Именно данный вид иерархии реализуется механизмом наследования объектно-ориентированного программирования.

Далее перейдем к определению блочно-иерархического подхода. Будучи в значительной степени отражением природных и технических систем, программные системы обычно являются иерархическими, так как обладают описанными выше свойствами. На этих свойствах иерархических систем строится блочно-иерархический подход. К их исследованию или созданию мы и приступаем. Этот подход предполагает сначала создавать части таких объектов (блоки, модули), а затем собирать из них сам объект [2].

Также мы должны дать следующее определение: процесс разбиения сложного объекта на сравнительно независимые части получил название декомпозиции. При декомпозиции учитывают, что связи между отдельными частями должны быть слабее, чем связи элементов внутри частей. Кро-

ме того, чтобы из полученных частей можно было собрать разрабатываемый объект, в процессе декомпозиции необходимо определить все виды связей между собой.

Также стоит упомянуть, что при создании очень сложных объектов процесс декомпозиции выполняется многократно: каждый блок, в свою очередь, декомпозируют на части, пока не получают блоки, которые сравнительно легко разработать. Данный метод разработки получил название «пошаговой декомпозиции».

В процессе декомпозиции необходимо выделить аналогичные блоки, которые можно было бы разрабатывать на общей основе.

Таким образом, как уже упоминалось выше, разбиение на части обеспечивает увеличение степени повторяемости кодов и, соответственно, снижение стоимости разработки.

Теперь мы подошли к определению понятия «иерархия». Необходимо уметь составлять схему соотношения абстрактного и конкретного в описании блоков при блочно-иерархическом подходе, о котором будет сказано ниже. Итак, результат декомпозиции обычно представляют в виде схемы иерархии, на нижнем уровне которого обычно располагают сравнительно простые блоки, а на верхнем — объект, подлежащий разработке. На каждом иерархическом уровне описание блоков выполняют с определенной степенью детализации, абстрагируясь от несущественных деталей. Следовательно, для каждого уровня используют свои формы документации и свои модели, отражающие сущность процессов, выполняемых каждым блоком.

Здесь нужно сформулировать некоторое правило, необходимое для дальнейших рассуждений, а именно: для объекта в целом удастся сформулировать лишь самые общие требования, а блоки нижнего уровня должны быть специфицированы так, чтобы из них действительно можно было собрать работающий объект. Другими словами, чем больше блок, тем более абстрактным должно быть его описание (см. рис.).

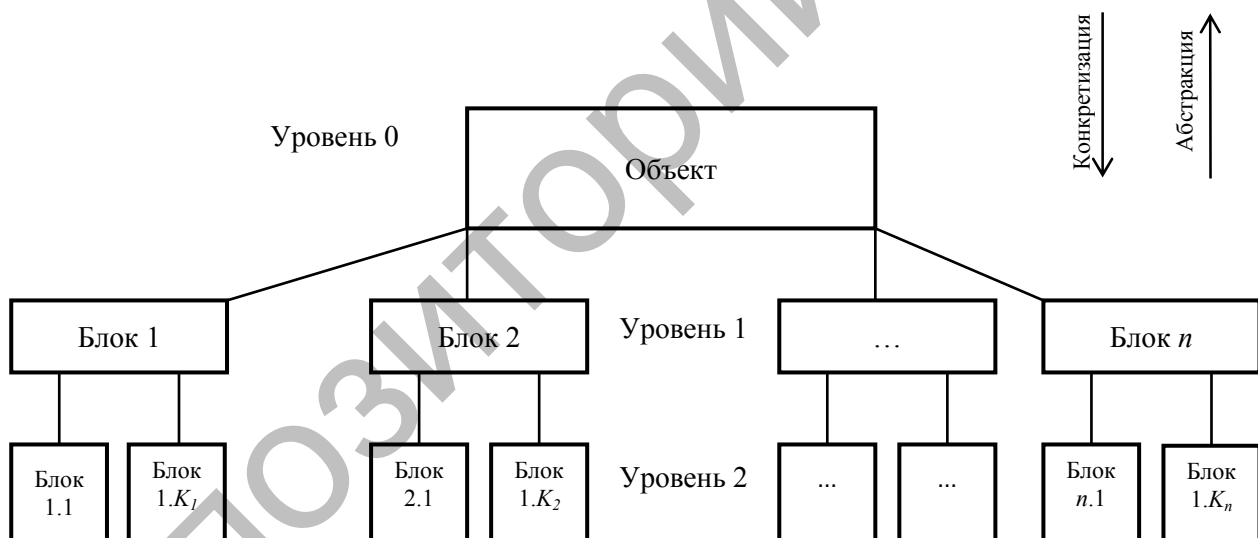


Рисунок. Соотношение абстрактного и конкретного в описании блоков при блочно-иерархическом подходе

Таким образом, в основе блочно-иерархического подхода лежат декомпозиция и иерархическое упорядочение. Необходимо отметить, что важную роль играют также следующие принципы:

- непротиворечивость — контроль согласованности элементов между собой;
- полнота — контроль на присутствие лишних элементов;
- формализация — строгость методического подхода;
- повторяемость — необходимость выделения одинаковых блоков для удешевления и ускорения разработки;
- локальная оптимизация — оптимизация в пределах уровня иерархии.

На основе сказанного выше можно ввести следующие два понятия. Первое — совокупность языков модулей, постановок задач, методов описаний некоторого иерархического уровня принято называть уровнем проектирования.

Второе — каждый объект в процессе проектирования, как правило, приходится рассматривать с нескольких сторон. Различные взгляды на объект принято называть аспектами проектирования.

Таким образом, резюмируя сказанное выше, сформулируем следующую парадигму проектирования: использование блочно-иерархического подхода применительно к программным системам стало возможно только после конкретизации общих положений подхода и внесения некоторых изменений в процесс проектирования. При этом структурный подход учитывает только свойства иерархии «целое-часть», а объектный — использует еще и свойство иерархии «простое — сложное».

Теперь, после того как мы ввели блочно-иерархический подход, мы подошли к понятию процесса проектирования. Итак, в основе любого процесса проектирования сложного программного комплекса лежит метод декомпозиции (его разбиения на более простые составные части — компоненты, модули).

Здесь нужно привести основные этапы процесса проектирования, а именно:

1. Проектирование архитектуры программного комплекса.
2. Разработка внешних спецификаций для выделенных на первом этапе компонентов.
3. Проектирование структур компонентов.
4. Разработка спецификаций для структурных единиц (подпрограмм, классов, модулей), выделенных в каждом компоненте на третьем этапе проектирования.
5. Проектирование структур данных и алгоритмов для выделенных в каждом компоненте структурных единиц.

Далее необходимо проанализировать приведенные выше этапы. Первые два этапа не являются обязательными и выполняются только при проектировании больших программных систем, которые могут быть разбиты на относительно независимые друг от друга компоненты (например, система программирования может быть разбита на текстовый редактор, подсистему управления файлами, компилятор, отладчик, справочную систему и т.п.).

Следующий, третий, этап необходимо рассмотреть отдельно, так как он является ключевым для процесса проектирования. Таким образом, на этапе проектирования архитектуры программного комплекса определяются:

- функции, выполняемые каждым компонентом;
- точные и однозначные сопряжения (интерфейсы) между компонентами;
- структура передач управления между компонентами;
- структура потоков данных;
- иерархическая структура асинхронно (параллельно) выполняющихся процессов (если такое предусмотрено);
- структура распределения оперативной памяти между компонентами;
- структура использования компонентами разделяемых устройств (например, внешних коммуникаций).

Результаты проектирования архитектуры программного комплекса отражаются во внешних спецификациях его компонентов. Нужно отслеживать переход от одного этапа к другому, а также обращать внимание на проектирование структур компонентов программного комплекса. Цель — определение всех составных частей компонента (будем называть их структурными единицами), их иерархии и интерфейсов между ними. Результаты выполнения должны выражаться в виде спецификаций свойств структурных единиц, на основании которых будет проводиться проектирование структур и алгоритма работы.

#### Список литературы

- 1 Шаяхметова Б.К. Пути повышения качества создания программного обеспечения для информационных специальностей университета // Классический университет в парадигме современных знаний: Материалы междунар. науч.-практ. конф. — Караганда: Изд-во КарГУ, 2012. — С. 376–378.
- 2 Хорев П.Б. Объектно-ориентированное программирование: Учеб. пособие для студентов учреждений высш. проф. образования. — 3-е изд., испр. — М: Изд. центр «Академия», 2011. — 448 с.

Б.К.Шаяхметова, Р.А.Сыздықова, К.С.Шәукенова, Ю.Н.Антипов

**Бағдарламаны жобалау сұрақтары**  
**блокты-сатылы амал, есептің міндеттерін құру**

Күрделі жүйелерге бағдарламалық нәтижелерді құру үшін, оны, біріншіден, ұсақтап бөлу (декомпозиция процесі) қажет, ал кейін есептеп, міндеттің қою керек. Мысалы, табиғатта және техникада көпшілік күрделі жүйелер ішкі сатылы құрылымдардан тұрады. Мақалада бағдарламалық жасақтаманы құруда қолданылатын әдістемелер, блоктық-сатылы амалдар қарастырылған, есептеу міндеттердің негізі тәсілдері тұжырымдалып, күрделі бағдарламаның жиындық декомпозициялық процесі құралған.

B.K.Shayakhmetova, R.A.Syzdykova, K.S.Shaukenova, Yu.N.Antipov

**Questions about program design**  
**block-hierarchical approach, setting objectives**

To create a complex software system, in the first place is to divide it into smaller parts (decomposition process), and then make a formulation of the problem. The vast majority of complex systems, both in nature and in technology have a hierarchical internal structure. The article highlights the techniques that involved in creating software, block-hierarchical approach are considered, the basic situation of formulation of the problem are given, the process of decomposition of complex software systems are mentioned.

References

- 1 Shayakhmetova B.K. *Classical university in the paradigm of modern knowledge: Proceedings of the International Scientific and Practical Conference*, Karaganda: Publ. of KSU, 2012, p. 376–378.
- 2 Horev P.B. *Object-oriented programming. The manual for students of higher education institutions* / P.B.Horev, 3rd pub., rev., Moscow: Publ. Center «Academiya», 2011, 448 p.