

Мякишев Д.В. Многомерная метамодель архитектуры программного обеспечения систем управления технологическими объектами. // Проблемы информатики в образовании, управлении, экономике и технике: Сб. статей XVI Междунар. научно-техн. конф. – Пенза: ПДЗ, 2016. – С. 133-138.

УДК004.415.2.031.43

МНОГОМЕРНАЯ МЕТАМОДЕЛЬ АРХИТЕКТУРЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СИСТЕМ УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМИ ОБЪЕКТАМИ

Д.В. Мякишев

MULTIDIMENSIONAL METAMODEL OF I&C SYSTEMS SOFTWARE ARCHITECTURE

D.V. Myakishev

Аннотация. В статье описывается метамодель архитектуры ПО, созданная на основе многомерного представления программной системы с разных точек зрения. Предлагаемая метамодель обеспечивает разработчиков компактным представлением знаний об особенностях построения ПО систем управления, позволяет систематизировать процессы проектирования, придать им регулярный формализованный характер.

Ключевые слова: архитектура программного обеспечения, декомпозиция, объектно-ориентированное проектирование, управление технологическими объектами.

Abstract. This article is describing a metamodel of the software systems architecture, based on multidimensional representation of a software system from various perspectives. This metamodel provides developers with compact knowledge about I&C systems software distinctive features, allows to organize design process, to give it ordered and formalized manner.

Keywords: software architecture, decomposition, object-oriented design, technical means control.

При создании сложного программного обеспечения (ПО), в частности ПО систем управления технологическими объектами (ТО), важную роль играет архитектура ПО, образующая концептуальную основу проекта. В современной программной инженерии понятие «программная архитектура» определяется как: «набор существенных решений относительно: организации программной системы; выбора структурных элементов, составляющих систему, и их интерфейсов; поведения этих элементов, определенного в их кооперациях; объединения этих структурных и поведенческих элементов в более крупные подсистемы; архитектурного стиля, определяющего организацию системы: статические и динамические элементы и их интерфейсы, кооперацию и композицию» [1].

Главным процессом создания архитектуры ПО является декомпозиция – процесс разделения целого на составные части, создания структуры проектируемого объекта. Как правило, при проектировании декомпозиция применяется многократно, до тех пор, пока в ее результате мы не получим набор простых, относительно их реализации, составных частей (компонентов), необходимый и достаточный для дальнейшего детального проектирования.

В первую очередь архитектура определяет структуру программной системы – совокупность ее функциональных элементов и связей между ними. В зависимости от того, какая форма (рабочая абстракция) была выбрана для предоставле-

ния структурных элементов системы, такая и будет структура (модульная, объектная и т. д.). Современный взгляд на архитектуру сложной программной системы заключается в том, что для ее представления недостаточно структуры одного типа [2]. Многомерное представление системы образуется совокупностью нескольких структур, каждая из которых отражает свой системный аспект.

Ранее автором был предложен подход к проектированию ПО систем управления на базе встраиваемых ЭВМ, основанный на применении в качестве концептуальной основы проекта базовой структуры – метамодели создаваемой системы [3]. Базовая структура (БС) позволяет в каждом конкретном проекте упорядочить процессы декомпозиции, придать им регулярный планомерный характер, сформулировать критерии или правила декомпозиции, что очень важно для качественного проектирования.

Для создания БС выполняется декомпозиция исходного представления ПО в трех аспектах: функциональном, операционном и программно-техническом. Результатом каждой декомпозиции является соответствующая структура. Далее производится объединение этих трех структур в единую метамодель путем связывания их элементов (задач, подзадач, процессов и модулей) бинарными отношениями.

В настоящее время несомненным преимуществом обладает парадигма объектно-ориентированного программирования (ООП), в рамках которой создается большая часть современного ПО. Однако прямое применение ООП при создании сложных программных систем вызывает определенные затруднения, в частности, при решении основной задачи ООП – определения набора классов, адекватного создаваемой системе. Поэтому на практике для решения данной задачи целесообразно использовать некоторые промежуточные представления (рабочие абстракции), обеспечивающие плавный переход от требований технического задания к объектам и соответственно классам.

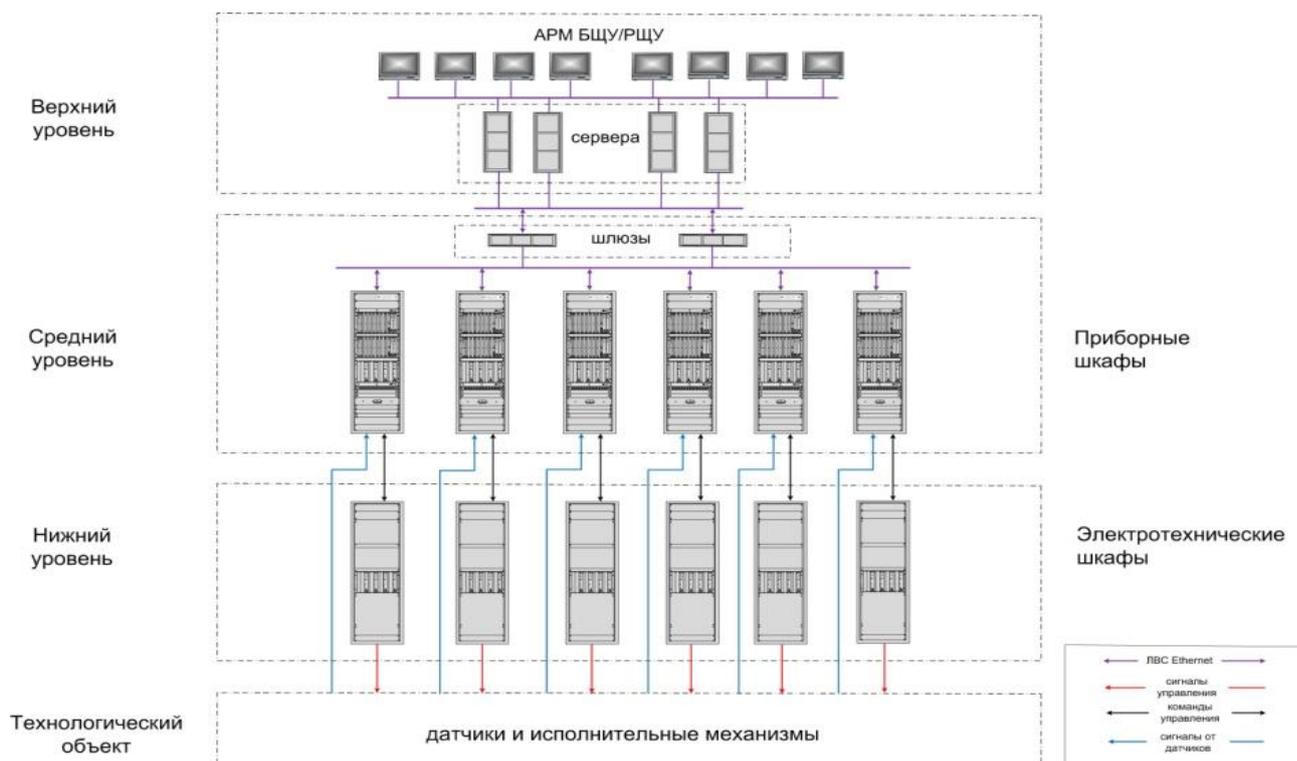


Рис. 1

Предлагается при проектировании ПО систем управления ТО использовать обновленную объектно-ориентированную БС – своеобразный симбиоз базовой структуры, предложенной автором [3], и модели «4+1» Филиппа Кратчена [2]. При этом для создания обновленной базовой структуры применяется сочетание функциональной, модульной и объектной декомпозиции.

Прежде чем перейти непосредственно к описанию обновленной БС (ОБС), необходимо кратко рассмотреть «матчасть» – аппаратную платформу, на которой должно функционировать создаваемое ПО.

Традиционно системы управления технологическими объектами разделяются на три уровня (рис. 1), каждый из которых образован своим характерным набором компонентов.

В данном случае нас интересуют основные «рабочие лошадки» системы – приборные шкафы или шкафы управления (ШУ). Они предназначены для решения задач сбора данных о технологическом объекте (ТО) от датчиков, их обработки и управления агрегатами ТО – исполнительными механизмами (ИМ). Количество ШУ в системе определяется объемом ТО – количеством технологических параметров и ИМ.

ШУ конструктивно объединяют в себе крейты с установленными модулями автоматизации (МА), средства коммутации, средства электропитания и защиты от помех. МА являются важнейшей составной частью комплекса. Именно МА являются основным строительным материалом, теми «кирпичиками», из которых строится система. Программные средства устанавливаются на МА, имеющие в своем составе микропроцессоры или микроконтроллеры.

В предлагаемой метамоделе архитектура ПО представляется объединением шести структур, каждая из которых отражает определенный ракурс (точку зрения) на внутреннее строение системы (рис. 2).

Система

Представляет проектируемое ПО как единое целое. Данное представление создается на основе «трех китов»:

- проекта системы управления, разработанный проектной организацией;
- «внутреннего проекта» – проекта предприятия-разработчика, обеспечивающего «привязку» внешнего проекта к аппаратным средствам автоматизации (шкафам управления, контроллерам, модулям);
- технического задания на разработку ПО, содержащего общие требования к создаваемому ПО, этапы работы и сроки их выполнения.

Подсистема

Компонент «подсистема» означает ПО, локализованное в одном ШУ. Декомпозиция на подсистемы имеет чисто физическую основу. Каждая подсистема объединяет задачи, размещаемые в МА одного ШУ.

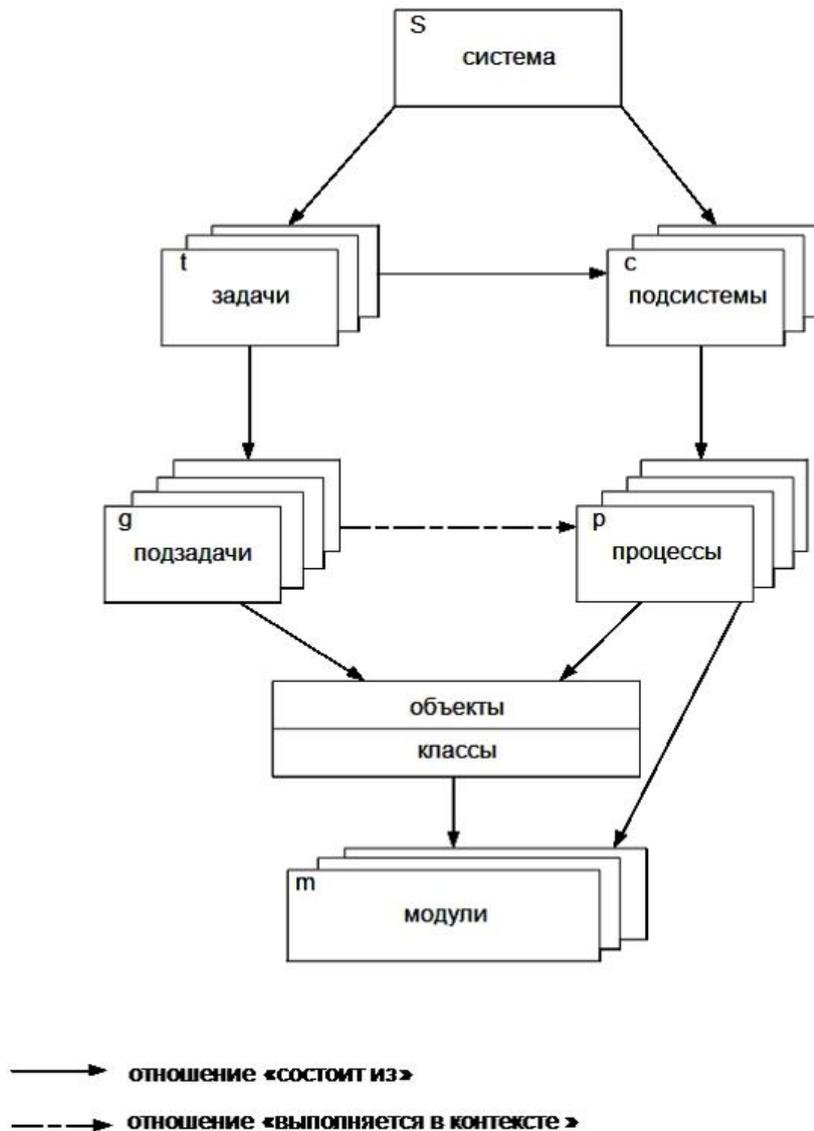


Рис. 2

Задача

Представляет результат вертикального деления (декомпозиции) системы. Задача – часть системы, выполняющая определенную системную функцию.

Задачи не пересекаются по выходам, т.е. результат выполнения конкретной задачи уникален. Возможно пересечение задач по входам. Выход одной задачи может являться входом для другой.

Каждая задача представляет собой цепочку взаимодействующих подзадач, причем каждая подзадача выполняется в рамках определенного процесса.

Каждая задача размещается в рамках одного ШУ.

Процесс

Каждый процесс – это часть системы, функционирующая параллельно (или квазипараллельно) с другими такими же частями – процессами. Каждое действие системы (системная функция) выполняется в рамках (контексте) какого-либо процесса. Процессы в системе взаимодействуют между собой.

Процессы распределяются между аппаратными программно-управляемыми средствами системы (модулями и блоками), имеющими в своем составе микропроцессоры или микроконтроллеры.

Различают «тяжелые» и «легкие» процессы. «Тяжелый» процесс выполняется в рамках собственного контекста, отличного от контекста других «тяжелых» процессов. Каждый «тяжелый» процесс привязан к «своему» модулю. «Легкий» процесс (другие названия – «поток», «нить») выполняется в рамках контекста своего тяжелого процесса.

Большинство процессов системы имеют статус «демонов», т.е. они выполняются в бесконечном цикле до их принудительного завершения.

Подзадача

Подзадача – часть задачи, выполняющаяся в рамках (контексте) определенного процесса. Каждая подзадача имеет статус «легкого» процесса.

Объект-класс

Объект – это рабочая абстракция в классическом понимании ОПП. Каждый объект принадлежит определенному классу. Каждая задача – это цепочка взаимодействующих объектов. Каждый объект выполняется в контексте определенного процесса. В ходе проектирования происходит постепенный переход от понятия «объект» к понятию «класс» в результате генерализации (обобщения).

Модуль

Модуль – это хранилище программного кода, независимая единица компиляции, компоновки и хранения.

Различают главные модули и библиотечные модули.

Главные модули – модули, содержащие функцию main (). Для каждого «тяжелого» процесса существует свой главный модуль.

Библиотечные модули содержат программный код классов, необходимых для программирования подзадач и процессов.

Таким образом, «в лице» ОБС мы получили многомерное представление создаваемой программной системы с разных точек зрения, которое обеспечивает компактное представление знаний об особенностях построения ПО определенного класса, позволяет систематизировать процессы проектирования, придать им регулярный формализованный характер.

Литература

1. Буч Г., Рамбо Д., Якобсон И. Язык UML. Руководство пользователя. 2-е изд. / пер. с англ. Н. Мухина. М.: ДМК Пресс, 2006. 496 с.
2. Philippe Kruchten. Architectural Blueprints. The «4+1» View Model of Software Architecture, Rational Software Corp. URL: <http://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf>
3. Мякишев Д.В. Разработка методов логического проектирования программного обеспечения систем управления динамическими объектами на базе встраиваемых ЭВМ: автореф. дис. канд. техн. наук : 05.13.11 / Моск. энерг. ин-т. М., 1991. 19 с.; Математическое и программное обеспечение вычислительных машин, комплексов, систем и сетей. URL: <http://dlib.rsl.ru/01000094703>

Мякишев Дмитрий Владимирович

г. Пенза, Россия

E-mail: dvm1948@yandex.ru

Myakishev D.V.

Penza, Russia