

# ПРИМЕНЕНИЕ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПОДХОДА К ПРОЕКТИРОВАНИЮ ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫХ СИСТЕМ

И.В.Брякин, Н.М. Лыченко

Институт автоматике и информационных технологий НАН КР, Кыргызстан,  
bivas2006@yandex.ru, nlychenko@mail.ru

Объектно-ориентированные технологии возникли в моделировании [1], потом получили мощное развитие в программировании, а в последнее время появляются публикации о применении объектно-ориентированного анализа и проектирования (ООАП) при разработке технических систем самого различного назначения. Стоит заметить, что инженерное проектирование всегда основывалось на интуитивном использовании объектно-ориентированной технологии. Однако нечеткая формализация процесса проектирования приводила к снижению его эффективности. Процесс проектирования технической системы трудно формализовать и обычно он занимает много времени и сил. Одним из способов преодоления этих трудностей является попытка придерживаться объектно-ориентированного подхода в проектировании и использования в качестве формализованного языка проектирования язык *UML*<sup>1</sup>. Являясь простым и мощным средством моделирования, *UML* может быть эффективно использован для построения концептуальных, логических и графических моделей сложных систем самого различного целевого назначения.

Учитывая современную направленность в разработке систем, связанную с объектно-ориентированным анализом и проектированием, представляется привлекательным использовать подход ООАП к проектированию ИИС.

**Суть ООАП.** Основная идея ООАП [2] состоит в рассмотрении предметной области и логического решения задачи с точки зрения *объектов* (понятий или сущностей). В процессе ООАП основное внимание уделяется определению и описанию объектов (или понятий) в терминах предметной области с последующим распределением обязанностей между ними. На основе выделения понятий предметной области строится *модель проектируемой системы*.

Известно, что основная стратегия борьбы с проблемами, связанными со сложностью системы (проблемы) – ее декомпозиция, т.е. разделение системы (проблемы) на мелкие управляемые элементы. До появления объектно-ориентированного подхода к анализу и проектированию наиболее популярной методологией декомпозиции являлись структурный анализ и проектирование. Этот подход заключается в декомпозиции задачи на функции или процессы приводящей к созданию иерархии процессов и подпроцессов. В рамках ООАП декомпозиция задачи выполняется в пространстве *объектов*, а не функций. В результате любая система представляется в виде группы объектов, общающихся друг с другом. Объекты одного типа могут быть объединены в *классы* (конкретное проявление класса – экземпляр класса, синоним объекта). Класс представляет собой некую сущность, которая задает некоторое общее поведение для объектов. *Цель проектирования* – распределение обязанностей между объектами для удовлетворения всех требований решаемой задачи.

**Обоснование применения объектно-ориентированного подхода при проектировании ИИС.** Для определения стратегии проектирования необходимо выяснить требования к проектируемой ИИС. В нотациях *UML* это можно сделать, построив диаграмму вариантов использования (см. Рис.1).

---

<sup>1</sup> Язык *UML (Unified Modelling Language)* - общецелевой язык визуального моделирования, который представляет собой систему обозначений, базирующуюся на диаграммах, и предназначен для спецификации, визуализации, проектирования и документирования компонентов программного обеспечения, бизнес-процессов и других систем [2].



Рис.1. Диаграмма вариантов использования ИИС.

Как видно из диаграммы, ИИС, в общем, должна удовлетворять различным функциональным требованиям [3]. Отсюда следует, что при проектировании системы разумно выделить для каждой функциональности общие элементы, соответствующие понятиям: измерение, вычисление, передача информации, преобразование информации и т.д. То есть, необходимо произвести декомпозицию системы на объекты, *различные сочетания которых позволят наделять систему различной функциональностью* в зависимости от требований практики и *получать качественно новые свойства системы* (быстродействие, надежность, унифицированность).

**Объектная декомпозиция ИИС.** Объектная декомпозиция основывается на принципах, используемых в структурной лингвистике: абстрагирование, инкапсуляция, модульность и иерархическая организация [4]. Эти принципы абсолютно применимы и для объектно-ориентированного представления ИИС. *Абстракция* выделяет внешнее поведение объекта и фиксирует основные характеристики объекта, которые отличают его от других видов объектов. Абстракцию обычно строят путем выделения обязанностей объекта. *Обязанности* означают обязательства объекта обеспечить определенное поведение. Пример: объект – чувствительный элемент, его обязанности: преобразовывать одну физическую величину (например, давление) в другую (например, перемещение). *Инкапсуляция* содержит и скрывает реализацию, которая обеспечивает поведение объекта. Инкапсуляция достигается с помощью информационной закрытости. Пример: при выборе чувствительного элемента проектировщику ИИС совсем не обязательно знать его внутреннюю структуру, как, впрочем, не обязательно ее знать и другим объектам ИИС. Важна только функциональность объекта, его интерфейс. Такой подход позволяет ограничить распространение изменений в одном из компонентов системы на всю систему.

*Модульность* предполагает, что в системе можно выделить взаимосвязанные группы абстракций, которые проектируются и изменяются достаточно независимо. И, наконец, *иерархическая организация* - это формирование из абстракций иерархической структуры. В результате определения иерархии сложная система становится обозримой человеком.

Основываясь на указанных принципах, в ИИС можно выделить, например, следующие объекты: чувствительный элемент, первичный измерительный преобразователь, средства квантования и дискретизации, устройство аналого-цифрового преобразования, канал связи, пакет данных, информационно-вычислительный компонент и т.п.

Объекты подвергаются воздействию или сами воздействуют на другие объекты, реализуя таким образом поведение системы. Поэтому следующая задача этапа анализа –

*определение связей (ассоциаций) между понятиями-объектами.* Отношения между парой объектов основываются на взаимной информации о разрешенных операциях и ожидаемом поведении. В ИИС особо интересны два вида отношений между объектами: связи и агрегация. Связь — это физическое или понятийное соединение между объектами (например, возможный путь для передачи сообщений). Пример: Чувствительный Элемент передает физическую величину Первичному Преобразователю. Агрегация обозначает отношения объектов в иерархии «целое/часть». Пример: Чувствительный Элемент и Первичный Преобразователь – составные части Первичного Измерительного Преобразователя.

**Концептуальная модель ИИС.** При проектировании ИИС (подобно проектированию программных систем [5]) понятия предметной области можно представить в виде *классов*, а их конкретные проявления - в виде объектов. Однако программные классы с их набором методов необходимо заменить «классами-устройствами» с набором поведений, свойственных данному объекту. Таким образом, концептуальную модель ИИС можно представить в виде диаграммы классов (Рис. 2). На этой диаграмме изображены классы и отношения между ними. Графически класс изображается в виде прямоугольника, включающего секции с именем, свойствами (атрибутами) и операциями (обязанностями). Классы в ИИС можно сгруппировать в группы-компоненты.

Первый компонент – *измерительный* – образуют классы *ПИП* и *ВИП*. Класс Первичный Измерительный Преобразователь (*ПИП*) – агрегат, чьи экземпляры включают по одному экземпляру классов *Чувствительный Элемент* (преобразовывает одну физическую величину (ФВ1) в другую (ФВ2)) и *Первичный преобразователь* (преобразовывает ФВ2 в электрический сигнал). Экземпляры Чувствительный Элемент и Первичный Преобразователь включаются в агрегат физически (на диаграмме это показано с помощью отношения *композиция*). Операция (обязанность) класса ПИП – получить первичную измерительную информацию (ПИИ). Операция Вторичного Измерительного Преобразователя (*ВИП*) - получить из ПИИ вторичную измерительную информацию, т.е. в определенном смысле выполнить функцию ее нормирования. ВИП может быть конкретно реализован в виде различных схем: дифференциальной, интегральной, логарифмической, компенсирующей и пр. На диаграмме классов класс ВИП удобно представить как родоначальный класс (шаблон) различных конкретных классов (Дифференциатор, Компенсатор, Усилитель и другие), что показано с помощью отношения *конкретизация*.

Второй компонент – *связующий* – образует один класс *Канал связи*. Он является родительским классом по отношению к четырем классам: Проводной Аналоговый Канал (*ПАК*), Беспроводной Аналоговый Канал (*БАК*), Цифровая Линия Длинной Связи (*ЦЛДС*), Цифровая Линия Короткой Связи (*ЦЛКС*). Операции этих классов – передавать сигнал - определены родительским классом, однако каждый из указанных потомков класса *Канал связи* предоставляет свой метод реализации этой операции. Отношение между родительским классом и классами-потомками – *наследование*. Символ {xor}, расположенный возле пунктирной линии, соединяющей два элемента, указывает на ограничение: сигнал может передаваться лишь по одному из указанных каналов.

Третий компонент – *информационно-подготовительный* - состоит из класса-шаблона *Средства Квантования и Дискретизации*, класса-ассоциации *Протокол Передачи Данных* и класса *Пакет Данных*. Класс-шаблон *Средства Квантования и Дискретизации* преобразует аналоговый сигнал в цифровой и порождает конкретные классы: *Специализированный АЦП* на базе микроконтроллеров, *Универсальный АЦП* (на базе универсальной платы РС11710, например), *Аналого-Частотное Преобразование (АЧП)*, *Аналого-Временное Преобразование (АВП)* и другие. Ассоциация, отображающая структурные отношения между экземплярами классов *Средства Квантования и Дискретизации* и *Пакет Данных* обладает свойствами класса, поэтому к линии ассоциации пунктирной линией присоединен класс-ассоциация *Пакет Передачи Данных*. Он, в свою очередь, представляет собой класс-шаблон, который

связывается с помощью отношения *конкретизации* с классами (*RS232, RS458, USB, LPT, PCI*), реализующими различные протоколы передачи данных.

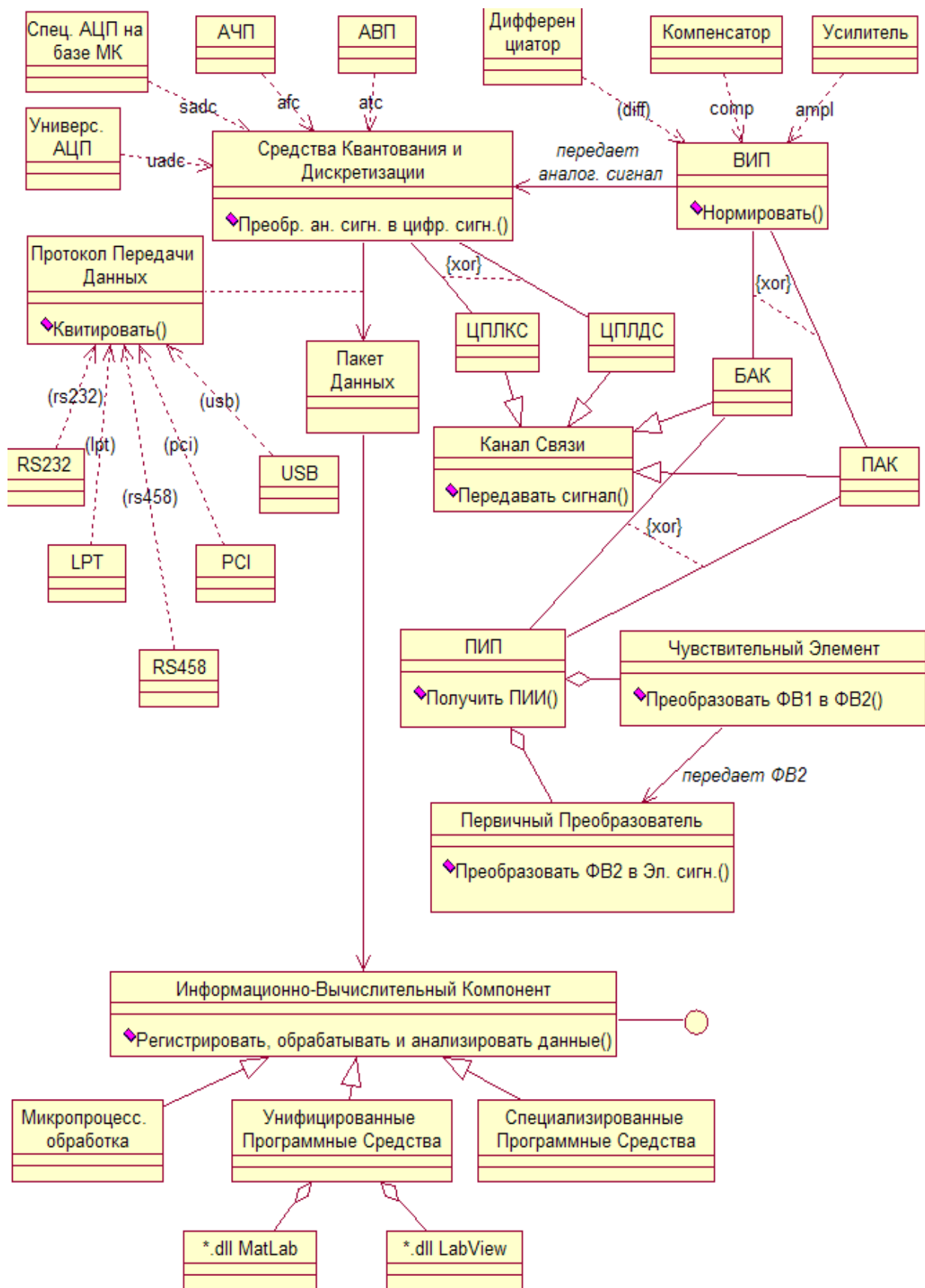


Рис. 2. Концептуальная модель ИИС.

Четвертый компонент – *информационно-вычислительный* – представляется одним метаклассом, назначение которого – регистрировать, обрабатывать и анализировать данные. Этот класс может быть представлен как родительский класс по отношению к трем классам: *Микропроцессорная Обработка*, *Универсальные Программные Средства* (включает в себя библиотеки *MatLab* и *LabView*), *Специализированные Программные Средства*. Отношение между родительским классом и классами-потомками – *наследование*. Кроме того, на диаграмме обозначен интерфейсный класс, который осуществляет взаимодействие с пользователем через консоль.

Разработав концептуальную модель ИИС в виде диаграммы классов, можно перейти к разработке других диаграмм, позволяющих детализировать поведение объектов, и которые в совокупности дадут полное представление о проектируемой системе. Для ИИС это, прежде всего, диаграммы, иллюстрирующие динамику системы: *диаграмма последовательности*, показывающая, в какой последовательности появляются объекты при выполнении той или иной операции и какой поток сообщений при этом возникает (иллюстрирует динамику системы); *диаграмма состояний* – показывает последовательность состояний, в которой может оказаться объект в зависимости от внешних событий; *диаграммы деятельности*, отражающие переходы, вызванные внутренними процессами. Кроме того, необходимо разработать модели реализации, обеспечивающие представление ИИС в физическом мире, рассматривая вопросы упаковки логических элементов в компоненты и размещения компонентов в аппаратных узлах. Эта модель представляется в виде *диаграммы развертывания*, показывающей схему расположения процессоров и устройств и маршруты передачи информации, и – для информационно-вычислительного компонента – *диаграммы компонентов* ПО. *Итеративная разработка всех указанных диаграмм в своей совокупности и будет составлять объектно-ориентированное проектирование ИИС.*

**Заключение.** Таким образом, стратегия проектирования, основанная на ООАП, в наибольшей степени соответствует проектированию ИИС, так как:

- в основе ООАП лежит модульная (объектная) структура, что позволяет разрабатывать ИИС с гибкой расширяемой структурой;
- происходит четкое разделение системы на части: измерительную, связывающую, информационно-вычислительную; изменение одной из частей не влечет перепроектирования всей системы;
- применение *UML* (в особенности диаграмм, иллюстрирующих динамику работы системы) для проектирования ИИС позволяет значительно упростить процесс разработки и представления документации в унифицированном виде.

Все указанные особенности позволяют достаточно быстро проектировать работоспособные варианты подсистем ИИС и запускать их в эксплуатацию, а саму стратегию проектирования, основанную на ООАП, принять в качестве базовой стратегии проектирования ИИС.

### Литература

1. Бенькович Е.С., Колесов Ю.Б., Сениченков Ю.Б. Практическое моделирование динамических систем // – СПб.: БХВ-Петербург, 2002.- 464с.
2. Ларман К. Применение UML и шаблонов проектирования. Издательский дом «Вильямс», 2004. – 489с.
3. Брякин И.В. Универсальная система контроля и диагностики для объектов и окружающей среды. Тез. Докл. межд. научн. практич. конф. «Перспективы развития научно-инновационной деятельности». – Бишкек, 2008.
4. Сумарокова Л.Н. Системный подход в современной лингвистике // Системный метод и современная наука. – Вып.5. – Новосибирск: Изд. НГУ, 1979.– С. 32–41.
5. Орлов С. А. Технологии разработки программного обеспечения. Уч. Пособие. – СПб.: Питер, 2003. 480 с.: ил.