

ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ МОДЕЛЬ СЛОЖНОЙ СИСТЕМЫ

М.П. Силич

Рассматривается метод моделирования сложных социально-экономических систем, основанный на объектно-ориентированном подходе. Метод является основой информационной технологии проектирования и позволяет формировать иерархическую модель сложной системы, содержащую множество вариантов реализации ее компонент, на базе типовых классов. Для выбора оптимальных вариантов компонент предлагается использовать инкапсулированные в модель различные процедуры принятия решений.

Введение

Проектирование социально-экономических систем для решения сложных многофакторных проблем в различных сферах человеческой деятельности – чрезвычайно сложная проблема. Примерами таких систем являются: региональная система энергосбережения и обеспечения энергетической безопасности; территориальная система ресурсообеспечения населения; учебно-научно-инновационный комплекс. Сложность проектирования подобных систем обусловлена сложностью самого проектируемого объекта, вызванной его многофункциональностью, необходимостью интегрирования в разнородную окружающую среду, множественностью целей отдельных компонент и требований, предъявляемых к системе. Облегчить процесс проектирования, обеспечить сокращение затрат при увеличении эффективности и качества проектируемой системы может информационная технология. Технология обеспечивает структуризацию и регламентирование процесса проектирования, автоматизацию используемых методов и процедур, в том числе методов построения моделей сложной системы и процедур принятия решений на модели, а также использование типовых проектных решений, что облегчает применение технологии для решения конкретных задач.

Центральное место в информационной технологии проектирования занимает методология моделирования проектируемой системы. В последнее время все большую популярность приобретают объектно-ориентированные методы моделирования благодаря предоставляемым ими таким важным преимуществам, как комплексирование декларативных знаний с процедурными, возможность сборки модели из готовых повторно используемых компонент, простота внесения изменений в проект и др. [1, 2].

В данной статье рассматривается объектно-ориентированная методология моде-

лирования, лежащая в основе информационной технологии проектирования сложных социально-экономических систем [3 – 5]. Главной особенностью предлагаемой методологии является возможность объединять различные методики системного анализа и инженерии знаний на базе декларативной модели, формируемой с использованием экспертных знаний, описывающих типовые свойства, структуры и закономерности отдельных классов систем.

Структура модели сложной системы

Объектно-ориентированная модель сложной системы включает в себя следующие основные виды моделей (рис.1):

- модель классов (ClassModel), содержащую множество классов (структурированных наборов информационных единиц – атрибутов и методов) для описания компонент системы, а также множество отношений между классами и методы работы с классами;
- модель объектов (ObjectModel), содержащую описание конкретной реализации компоненты системы или множества вариантов ее реализации в виде объекта (экземпляра класса) или совокупности объектов, а также методы работы с объектами и их совокупностями;
- модель компонент системы (SubSystemModel), содержащую структуру сложной системы в виде дерева подсистем и присоединенных элементов, множество взаимосвязей между компонентами системы, а также методы для работы со структурами;
- модель вариантов реализации компонент системы (SubSystemObjectModel), содержащую дерево наборов реализаций (вариантов реализаций) подсистем и присоединенных элементов, а также методы для работы с наборами реализаций.

Все модели взаимосвязаны. На базе модели классов формируется модель объектов. Модель классов используется также при

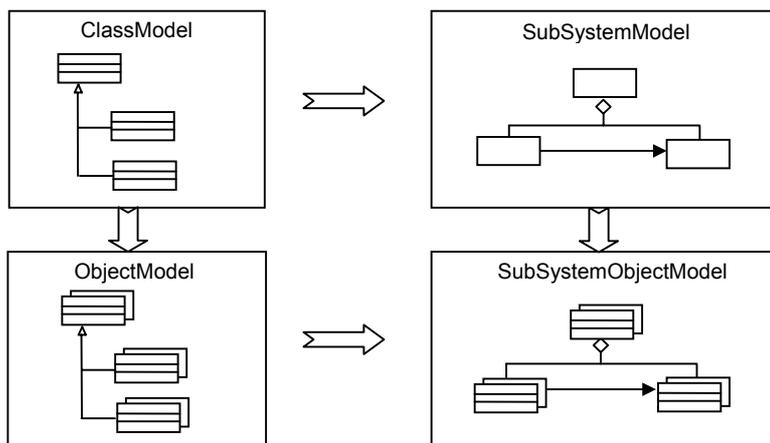


Рис. 1. Основные виды моделей сложной системы

формировании модели компонент. В свою очередь модель компонент и модель объектов используются при построении модели вариантов реализации компонент.

Рассмотрим подробнее каждую из перечисленных моделей.

Модель классов

Понятия класса и объекта являются основными понятиями любого объектно-ориентированного представления. В объектной модели сложной системы в виде объекта представляется любой компонент системы – сама система в целом, ее подсистемы и элементы, а также подсистемы и элементы окружающей среды. Объект представляет собой структурированное описание конкретного компонента. Классы же определяют типы используемых объектов и задают соответствующие структуры описания.

Модель классов M^C включает: множество классов $C = \{c_i\}$ для описания систем и их компонент, на котором определено отношение наследования (inheritance) $R^{in} \in C \times C$, а также множество методов (присоединенных процедур) P^C для работы с классами:

$$M^C = \langle C, R^{in}, P^C \rangle.$$

Класс представляет собой кортеж:

$$c_i = \langle n(c_i), \{n(c_j) \mid c_j R^{in} c_i\}, A(c_i), P(c_i) \rangle,$$

где $n(c_i)$ – имя класса; $\{n(c_j)\}$ – множество имен классов, от которых наследуется данный класс; $A(c_i) = \{a_m\}$ – множество атрибутов класса; $P(c_i)$ – множество методов класса.

Каждый из атрибутов задается тройкой:

$$a_m = \langle n(a_m), t(a_m), D(a_m) \rangle,$$

где $n(a_m)$ – имя атрибута; $t(a_m)$ – тип атрибута ($\langle \text{String} \rangle$, $\langle \text{Real} \rangle$, $\langle \text{Integer} \rangle$, $\langle \text{Class} \rangle$, ...); $D(a_m) = \{d_k\}$ – множество значений (домен) атрибута.

Домен задается в зависимости от типа атрибута. Например, для атрибута типа $\langle \text{Real} \rangle$ или $\langle \text{Integer} \rangle$ – в виде интервала, для атрибута типа $\langle \text{String} \rangle$ – в виде списка строк, для атрибута типа $\langle \text{Class} \rangle$ – в виде имени класса (списка имен классов). Посредством атрибутов типа $\langle \text{Class} \rangle$ в описание объекта может включаться описание другого объекта. Например, атрибуты подсистемы могут содержать ссылки на объекты, описывающие элементы данной подсистемы или некоторые комплексные свойства, которые сами описываются множеством собственных атрибутов.

Множество методов класса содержит обязательное подмножество методов доступа к значениям атрибутов, которые могут быть двух типов: get_value – метод получения значения атрибута; set_value – метод задания значения атрибута. Метод задания значения может быть представлен в виде множества правил-продукций, процедуры либо формулы, включающих арифметические и логические операции, ссылки на атрибуты текущего или другого класса, константы и др. Посредством методов доступа одни атрибуты могут определяться через значения других.

Множество методов для работы с классами P^C включает две основных группы методов: методы наследования, позволяющие создавать новые классы на базе имеющихся классов; методы обобщения, позволяющие автоматически или в диалоговом режиме формировать иерархию наследования классов.

ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ МОДЕЛЬ СЛОЖНОЙ СИСТЕМЫ

Модель объектов

На базе классов может быть создано множество объектов (реализаций классов) $O = \{o_k\}$. Каждый объект задается следующим образом:

$$o_k = \langle n(o_k), c(o_k), D(o_k) \rangle,$$

где $n(o_k)$ – имя объекта; $c(o_k)$ – указатель на класс, на базе которого реализован объект; $D(o_k) = \{d(a_m) \in D(a_m) \mid a_m \in A(c(o_k))\}$ – множество значений атрибутов, полученных как результат выполнения методов задания значения: `set_value`. Множество значений может быть представлено как дизъюнкция значений с различными факторами уверенности:

$$d_1(a_m) / cf(d_1(a_m)) \vee d_2(a_m) / cf(d_2(a_m)) \vee K,$$

где $cf(d_k(a_m)) \in [0,1]$ – фактор уверенности в значении атрибута, принимающий значение в интервале от 0 (полная недостоверность) до 1 (абсолютная достоверность). Фактор уверенности либо непосредственно задается разработчиком модели при вводе значения атрибута, либо определяется с помощью функции принадлежности, либо вычисляется с помощью формул нечеткой логики при выводе значения с помощью логических правил. Таким образом, каждый атрибут может одновременно принимать несколько значений с разными степенями уверенности.

Для отображения множества вариантов реализации некоторого класса, например, множества состояний системы в различные моменты времени, в различных точках пространства, в различных условиях и т.д., введем понятие мультиобъекта. Будем называть мультиобъектом множество объектов (набор реализаций) одного класса, выделенных в соответствии с некоторым признаком p :

$$O^p(c_i) = \{o_k \mid o_k = o(p), c(o_k) = c_i\} \subset O(c_i).$$

Каждому значению признака соответствует некоторая конкретная реализация, являющаяся, таким образом, функцией от p . Это означает, что значение атрибутов объекта может зависеть от значения признака и методы `set_value` атрибутов могут содержать в качестве параметров вычисления значения атрибута признак p . Различают три основных типа базовых признаков – время, пространство и группа (population). От них могут наследоваться конкретные признаки, причем признаки могут быть комбинированными, т.е. являться композицией нескольких признаков. Отдельный объект можно рассматривать, как вырожденный мультиобъект, мощность множества которого равна единице.

Модель объектов содержит множество мультиобъектов (реализаций различных

классов), а также методы для работы с мультиобъектами:

$$M^O = \langle (O^p_j(c_i)), P^O \rangle.$$

Множество P^O включает следующие основные группы методов: методы ввода, редактирования и удаления объектов; переборные методы генерации объектов; методы поиска объектов по заданному шаблону поиска; методы вывода закономерностей на множестве объектов; оценочные методы выбора оптимальных объектов.

Модель компонент

Термином «компонент» будем обозначать любую часть предметной области, которая может быть выделена и описана как некоторая самостоятельная сущность. Это и система в целом, рассматриваемая как «черный ящик» на верхнем уровне абстрагирования, и любая часть системы – подсистема, элемент, комплексное свойство.

Обозначим множество компонент через $K = \{k_i\}$. Компонент может быть представлен в виде кортежа: $k_i = \langle n(k_i), c(k_i) \rangle$, где $n(k_i)$ – имя компонента; $c(k_i)$ – класс, на который ссылается компонент. Обязательным является только имя компонента.

Из множества компонент выделим подмножество подсистем $S \subseteq K$, на котором определено отношение агрегации (aggregation)

$R^{ag} \in S \times S$. Это отношение типа «часть-целое». Оно устанавливается между подсистемами, одна из которых включает в качестве составной части другую. Множество подсистем, связанных отношениями агрегации, представляет собой дерево подсистем. Это стратифицированная иерархия, отражающая структуру проектируемой системы.

Дерево, как правило, формируется путем последовательной декомпозиции подсистем. Будем называть совокупность подсистем, полученных в результате декомпозиции одной и той же системы предыдущего уровня, подуровнем. При декомпозиции каждая подсистема порождает свой подуровень. Порождающую подсистему будем называть материнской, а подсистемы, составляющие подуровень – дочерними.

Формирование любого подуровня может осуществляться с помощью, так называемых, стандартных оснований декомпозиции. Основание декомпозиции можно представить, как объект, имеющий атрибут с именем материнской системы и атрибут, содержащий список имен дочерних подсистем, а также методы для порождения дочерних подсистем. Стан-

дартные основания декомпозиции могут храниться в библиотеке типовых моделей.

Среди множества компонент кроме подмножества подсистем будем выделять подмножество элементов $E \subseteq K$. При этом под элементом будем понимать активные и пассивные сущности, участвующие в деятельности подсистемы (например, конечные продукты, предметы деятельности, средства деятельности, исполнители) или комплексные свойства, описываемые набором характеристик (например, технологические параметры, экономические результаты, технические условия и т.д.).

Элемент может «присоединяться» к подсистеме или к другому элементу посредством указания его имени в качестве значения одного из атрибутов класса присоединяющего компонента. Таким образом, присоединяемый компонент дополняет объектное описание присоединяющего компонента, т.е. является частью его описания. Обозначим отношение присоединения (connection) через $R^{cn} \in K \times K$. На рис. 2 представлена структура взаимосвязи компонент сложной системы.

Помимо связей, отражающих структурные отношения (агрегации и присоединения), между подсистемами и между элементами могут устанавливаться разнообразные отношения, отражающие физические связи. Это, прежде всего, потоки, заключающиеся в передаче энергии, вещества или информации, пространственные, временные отношения, а также разнообразные эмпирические отношения. Назовем отношения данного вида отношениями ассоциации (association)

$R^{as} \in K \times K$. Чтобы избежать излишней сложности, будем учитывать только взаимосвязи

между подсистемами одного подуровня или между элементами одной подсистемы.

Таким образом, модель компонент системы включает в себя множество компонент системы, связанных отношениями агрегации, присоединения и ассоциации, а также методы для работы с компонентами:

$$M^K = \langle K, R^{ag}, R^{cn}, R^{as}, P^K \rangle.$$

Множество P^K включает следующие основные группы методов: методы создания и редактирования иерархии компонент, в том числе методы генерации подсистем подуровня на базе стандартных оснований декомпозиции; методы создания диаграмм взаимосвязей (ассоциации); методы вывода на диаграммах взаимосвязей.

Модель вариантов реализации компонент системы

Данная модель формируется на основе модели компонент путем сопоставления каждому компоненту его одной или нескольких реализаций, т.е. объекта или мультиобъекта. Если рассматривать объект как частный случай мультиобъекта, то модель вариантов реализации компонент системы может быть представлена в виде:

$$M^V = \langle \bigcup_l \{O_j^P(c_i) \mid c_i = c(k_l)\}, P^V \rangle.$$

В состав множества P^V должны быть включены методы, позволяющие согласовывать варианты реализации различных компонент между собой. К ним относятся, в частности методы координации.

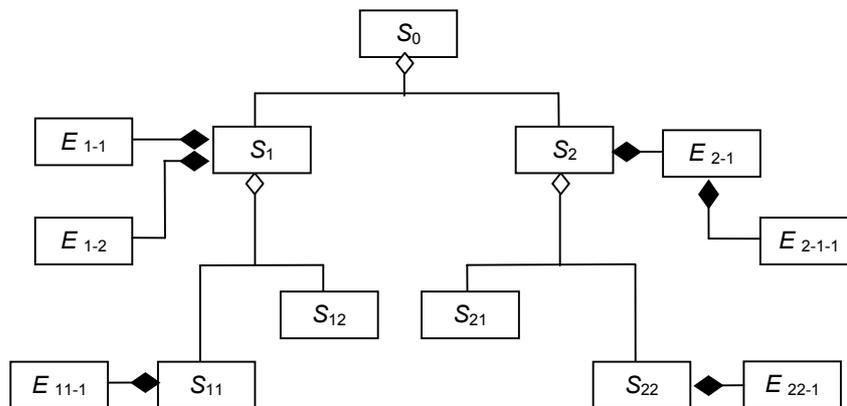


Рис. 2. Структура взаимосвязи компонент сложной системы

Поскольку компоненты сложной системы взаимосвязаны друг с другом, очевидно, что значения атрибутов, характеризующие некоторую реализацию одного компонента, могут зависеть от значений атрибутов, характеризующих реализации других компонент, связанных с ним. Другими словами, между атрибутами классов, сопоставленных различным компонентам сложной системы, могут существовать отношения зависимости. Поэтому в описание классов компонент необходимо включить соответствующие функциональные отношения.

Будем различать следующие типы межкомпонентных отношений зависимости:

1. Межуровневые зависимости, связывающие атрибуты материнской подсистемы s_0 с атрибутами дочерних подсистем или элементов. Как правило, данный вид зависимостей представляется в виде функциональных отношений, разрешенных относительно атрибутов материнской подсистемы:

$$a_0 = f(a_1, K, a_i, K),$$

$$a_0 \in A(c(s_0)), a_i \in A(c(k_i)), s_0 R^{ag} k_i \vee s_0 R^{cn} k_i.$$

Примером является зависимость затрат материнской системы от затрат ее подсистем, определяемая как сумма.

2. Внутриуровневые зависимости, отражающие взаимное влияние атрибутов подсистем, входящих в подуровень (межподсистемные зависимости - баланс входов-выходов подсистем), либо зависимости атрибутов элементов одной и той же подсистемы (межэлементные зависимости). Пример межподсистемных зависимостей: объем продукции, производимой некоторой подсистемой, должен быть равен объему продукции, поступающей на вход другой подсистемы, умноженному на коэффициент потерь, возникающих при передаче.

Методы координации позволяют использовать межкомпонентные зависимости для согласования оптимальных вариантов компонент сложной системы.

Процесс моделирования

Рассмотрим, как используется объектно-ориентированная модель на различных этапах проектирования (перепроектирования) сложной системы.

На подготовительном этапе осуществляется планирование проведения проектирования и организация работ. Поскольку процесс проектирования сложной системы сам может рассматриваться как система, то для

него тоже может быть построена объектно-ориентированная модель. Основу модели составляет дерево компонент, т.е. иерархия этапов и подэтапов процесса проектирования. Класс описания любого этапа содержит такие атрибуты, как «исходные данные», «результат», «исполнитель», «срок окончания», «стоимость» и др. Для каждого подуровня может быть составлена диаграмма взаимосвязей, отражающая связи между этапами. Затем формируются варианты реализации этапов и выбирается наиболее перспективная комбинация вариантов, согласованных по срокам, общей стоимости проекта и т.д.

Следующим этапом, как правило, является анализ, на котором исследуется существующая (перепроектируемая) система или аналоги проектируемой системы. Могут быть выделены отдельные подэтапы, соответствующие различным видам анализа – «ретроспективный анализ», «сравнительный анализ», «причинный анализ». При использовании любого вида анализа сначала формируется модель компонент исследуемой системы и составляются описания классов компонент. Для формирования иерархии подсистем могут быть использованы стандартные основания декомпозиции, а для составления описания классов – типовые описания, хранящиеся в библиотеке классов в виде иерархии наследования. Затем на базе классов создаются экземпляры описаний, т.е. объекты или мультиобъекты и осуществляется сравнение объектов с целью установления тенденций, выявления проблем и ограничений.

При ретроспективном анализе формируются мультиобъекты по временному признаку, отражающие состояния компонент в различные моменты времени. На основе сравнения объектов с помощью различных методов, например, методов статистической обработки, методов индуктивного вывода и обучения, могут быть выявлены закономерности в виде некоторых формул, правил или процедур, которые должны быть внесены в описание класса. В дальнейшем они могут быть использованы при выработке решений.

При сравнительном анализе формируются мультиобъекты по признаку «группа (population)», содержащие описания аналогов, а также «идеальных» вариантов компонент. Сравнение объектов может осуществляться по отдельным атрибутам, выступающим в качестве критериев. При этом могут быть использованы многокритериальные методы с использованием различных видов интегральных критериев, например «максимум

суммы взвешенных оценок», «минимум суммы отклонений от идеальной точки» и т.д.

Стратегия «прохождения» иерархии подсистем в процессе анализа может быть различной, однако более предпочтительна стратегия «снизу вверх», поскольку атрибуты подсистем верхних уровней не только непосредственно «снимаются» с исследуемой системы, но и определяются через значения атрибутов подсистем нижних уровней.

На этапе постановки целей формулируются цели проектируемой системы и ее основных подсистем. При этом в качестве отдельных компонент могут быть выделены цели (требования) систем окружающей среды (вышестоящих систем, потребителей, партнеров), а также различные аспекты целеполагания (политический, экономический, социальный, экологический). Словесные формулировки целей необходимо дополнить значениями критериев, выступающих индикаторами достижения целей. К каждому компоненту иерархии целей присоединяется мультиобъект, отражающий планируемые значения критериев в различные моменты времени прогнозируемого периода.

Основным содержанием этапа выработки решений является синтез вариантов реализации компонент системы и выбор наиболее перспективных вариантов. Сначала формируется иерархия компонент проектируемой системы и создаются описания классов компонент. При этом может быть использована модель компонент, созданная на этапе анализа.

Поиск оптимального варианта отдельных компоненты может осуществляться двумя способами. Первый способ «от средств – к цели» предполагает генерирование различных вариантов, их оценку и выбор оптимального варианта. Для генерации вариантов могут использоваться переборные методы, например, метод морфологического анализа. При использовании этого метода из множества атрибутов выбираются, так называемые, базовые атрибуты (конструктивные, управляемые параметры) и формируются различные комбинации значений этих атрибутов. Недопустимые комбинации отбрасываются, а оставшиеся оцениваются. Для выбора оптимальных объектов, удовлетворяющих заданным ограничениям и являющихся наилучшими в смысле выделенных критериев эффективности, могут быть использованы различные методы теории выбора.

Второй способ («от цели – к средствам») предполагает последовательное выстраива-

ние дерева решений на множестве атрибутов, формируемого от целевых атрибутов через промежуточные к базовым, определяющим варианты. Дерево решений может быть представлено либо в виде дерева «И/ИЛИ», либо в виде когнитивной карты (в том числе и нечеткой), либо в виде дерева анализа, построенного методом анализа иерархий Т.Саати [6].

Общая стратегия поиска решений на иерархической модели – «сверху – вниз». Сначала на верхнем уровне находится агрегированный вариант всей системы в целом в пространстве обобщенных атрибутов. Далее определяются оптимальные варианты подсистем следующего уровня и т.д. При этом на каждом уровне целесообразно определять оптимальное и непротиворечивое сочетание вариантов подсистем, входящих в отдельный подуровень. В качестве критерия эффективности подуровня, как правило, выбирается критерий материнской системы, для которого задана зависимость от локальных критериев подсистем в виде некоторой обобщенной функции. Оптимальным является такое сочетание вариантов подсистем, для которого выполняются межподсистемные и межуровневые ограничения и при этом критерий подуровня достигает своего наилучшего значения. Для нахождения оптимальных комбинаций вариантов могут использоваться процедуры координации.

СПИСОК ЛИТЕРАТУРЫ

1. Ойхман Е.Г., Попов Э.В. Реинжиниринг бизнеса: Реинжиниринг организаций и информационных технологии. – М.: Финансы и статистика, 1997. – 336 с.
2. Буч Г. Объектно-ориентированное проектирование с примерами применения: Пер. с англ. – М.: Конкорд, 1992. – 519 с.
3. Силич М.П. Системная технология: объектно-ориентированный подход. – Томск: Том. гос. ун-т систем управления и радиоэлектроники, 2002. – 224 с.
4. Силич В.А., Силич М.П. Проектирование сложной системы на основе объектно-ориентированного подхода. Известия Томского политехнического университета. – 2003. – Т. 306. – №2. – С. 99-103.
5. Силич В.А., Силич М.П. Метод объектного моделирования для проектирования сложных систем. Автоматизация и современные технологии. – 2003. – №4. – С. 14-21.
6. Саати Т., Кернс К. Аналитическое планирование. Организация систем: Пер. с англ. - М.: Радио и связь, 1991. - 224 с.