

Курс лекции по дисциплине YaPZ7302 Языки представления знаний

Лекция 15. Методы структурирования и формализации

Цель: Изучение методов структурирования и формализации

План:

1. Структурирование знаний
2. Проектирование сложных систем
3. Подходы к проектированию сложных систем
4. Объектно-структурный подход (ОСП)
5. Стратификация знаний
6. Алгоритм ОСА (объектно-структурного анализа)
7. Глобальный анализ
8. Анализ страт

Стадия концептуального анализа или структурирования знаний традиционно является (наряду со стадией извлечения) "узким местом" в жизненном цикле разработки интеллектуальных систем [Adeli, 1994]. Методология структурирования близка к современной теории больших систем [Гиг, 1981] или сложных систем [Courtois, 1985], где традиционно акцент делается на процессе проектирования [Bertalanffy, 1950; Boulding, 1956]. Большой вклад в эту теорию внесли классики объектно-ориентированного анализа [Буч, 1992].

Разработку интеллектуальных систем с уверенностью можно отнести к данному классу задач, поскольку они обладают основными признаками сложности (иерархия понятий, внутриэлементные и межэлементные связи и пр.). Аналогичные концепции, но связанные не с общесистемными исследованиями, а рассматривающие информационные процессы в системах, таких как связь и управление, положили начало кибернетике как самостоятельной науке [Винер, 1958; Эшби, 1959]. Позднее, в 1960-х гг. были получены новые результаты по развитию математической теории систем высокого уровня общности [Месарович, Такахара, 1978]. Существенный вклад в теорию

систем и основы структурирования внесли отечественные исследователи [Моисеев, 1981; Глушков, 1964; Ивахненко, 1971; Пospelов, 1986] и др.

Системный анализ тесно переплетается с теорией систем и включает совокупность методов, ориентированных на исследование и моделирование сложных систем — технических, экономических, экологических и т. п.

Проектирование сложных систем и методы структурирования информации традиционно использовали иерархический подход [Месарович, Такахара, 1978] как методологический прием расчленения формально описанной системы на уровни (или блоки, или модули). На высших уровнях иерархии используются наименее детализованные представления, отражающие только самые общие черты и особенности проектируемой системы. На следующих уровнях степень подробности возрастает, при этом система рассматривается не в целом, а отдельными блоками.

На каждом уровне вводятся свои представления о системе и элементах. Элемент k -го уровня является системой для $(k - 1)$ уровня. Продвижение от уровня к уровню имеет строгую направленность, определяемую стратегией проектирования — дедуктивную нисходящую "сверху вниз" (top-down) или индуктивную восходящую "снизу вверх" (bottom-up).

Предлагаемый ниже объектно-структурный подход позволяет объединить две, обычно противопоставляемые, стратегии проектирования. Синтез этих стратегий, а также включение возможности итеративных возвратов на предыдущие уровни обобщений позволили создать дуальную концепцию, предоставляющую аналитику широкие возможности на стадии структурирования знаний как для формирования концептуальной структуры предметной области Sk , так и для функциональной структуры Sf .

Рис. 21 иллюстрирует дуальную концепцию при проектировании Sf для ЭС помощи оператору энергетического блока

Нисходящая концепция (top-down) декларирует движение от $n \Rightarrow n + 1$, где n — n -й уровень иерархии понятий ПО (предметной области) с

последующей детализацией понятий, принадлежащих соответствующим уровням.

$$STR_{td} : P_i^n, \dots, P_i^{n+1} \Rightarrow P_{k_i}^{n+1}$$

где:

- n — номер уровня порождающего концепта;
- i — номер порождающего концепта;
- k_i — число порождающих концептов, сумма всех k_i по i

составляет общее число концептов на уровне $n + 1$.

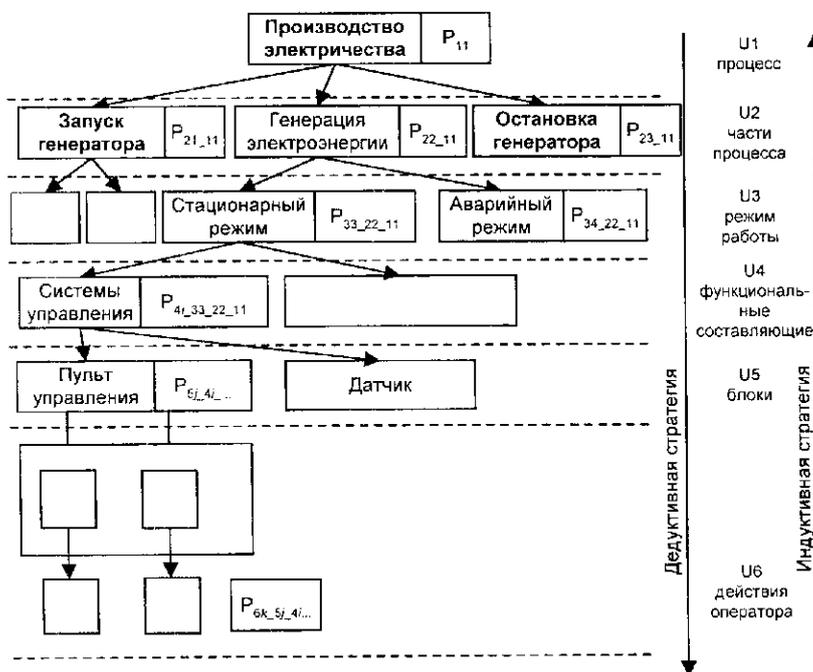


Рис. 21. Дуальная стратегия проектирования

Восходящая концепция (bottom-up) предписывает движение $n \Rightarrow n - 1$ с последовательным обобщением понятий.

$$STR_{bu} : P_i^n, \dots, P_{k_i}^{n+1} \Rightarrow P_i^{n-1}$$

где:

- n — номер уровня порождающих концептов;
- i — номер порождаемого концепта;
- k_i — число порождающих концептов, сумма всех k_i по i составляет

общее число концептов на уровне n .

Основанием для прекращения агрегирования и дезагрегирования является полное использование словаря терминов, которым пользуется эксперт, при этом число уровней является значимым фактором успешности структурирования.

В целом существующие подходы к проектированию сложных систем можно разделить на два больших класса:

- *структурный (системный) подход* или анализ, основанный на идее алгоритмической декомпозиции, где каждый модуль системы выполняет один из важнейших этапов общего процесса;
- *объектный подход*, связанный с декомпозицией и выделением не процессов, а объектов, при этом каждый объект рассматривается как экземпляр определенного класса.

В структурном анализе [Yourdon, 1989] разработано большое число выразительных средств для проектирования, в том числе графических [Буч, 1991]:

- диаграммы потоков данных (DFD, data-flow diagrams), структурированные словари (тезаурусы), языки спецификации систем, таблицы решений;
- стрелочные диаграммы "объект-связь" (ERD, entity-relationship diagrams), диаграммы переходов (состояний);
- деревья целей;
- блок-схемы алгоритмов (в нотации Насси— Шнейдермана, Фестля и др.);
- средства управления проектом (PERT-диаграммы, диаграммы Ганта и др.).

Множественность средств и их некоторая избыточность объясняется тем, что каждая предметная область, используя структурный подход как универсальное средство моделирования, вводила свою терминологию, наиболее подходящую для отражения специфики конкретной проблемы. Поскольку инженерия знаний имеет дело с широким классом ПО (это

"мягкие" ПО), встает задача разработки достаточно универсального языка структурирования.

Объектный (объектно-ориентированный) подход (ООП) был разработан в 1979 году [Jones, 1979], а затем развит в работах [Peters, 1981; Shaw, 1984; Буч, 1993].

ООП, возникший как технология программирования больших программных продуктов, базируется на основных элементарных понятиях [Буч, 1993]:

- объекты и классы как объекты, связанные общностью структуры и свойств;
- классификации как средства упорядочения знаний;
- иерархии с наследованием свойств;
- инкапсуляции как средства ограничения доступа;
- методы и полиморфизм для определения функций и отношений.

ООП имеет систему условных обозначений и предлагает набор моделей для проектирования сложных систем. Широкое распространение объектно-ориентированных языков программирования C++, CLOS, Smalltalk и др. успешно демонстрируют жизнеспособность и перспективность этого подхода. В последнее время этот подход успешно применяется и в CASE-средствах не только для проектирования программ, но и для моделирования бизнес-процессов. Особенную популярность приобретает язык UML (Unified Modelling Language) [Буч, Рамбо, Джекобсон, 2000], и программный инструментарий на нем основанный, например Rational Rose [Боггс, 2001].

Объектно-структурный подход (ОСП)

В качестве базисной парадигмы структурного анализа знаний и формирования поля знаний P_z можно предложить обобщенный объектно-структурный подход (ОСП) [Гаврилова, 1995].

Основные постулаты этой парадигмы заимствованы из ООП и расширены:

1. Системность (взаимосвязь между понятиями).

2. Абстрагирование (выявление существенных характеристик понятия, которые отличают его от других).
3. Иерархия (ранжирование на упорядоченные системы абстракций).
4. Типизация (выделение классов понятий с частичным наследованием свойств в подклассах).
5. Модульность (разбиение задачи на подзадачи или "возможные миры").
6. Наглядность и простота нотации.

Использование пятого постулата ОСП в инженерии знаний позволяет строить глобальные БЗ с возможностью выделить локальные задачи с помощью горизонтальных и вертикальных сечений (*см. ниже*) на отдельные модули пространства-описания предметной области.

Шестой постулат внесен в список последним, но не по значимости. В инженерии знаний формирование *PZ* традиционно является критической точкой [Гаврилова, Червинская, Яшин, 1988; Гаврилова, Червинская, 1992], т. к. создаваемая неформальная модель предметной области должна быть предельно ясной и лаконичной. Традиционно языком инженерии знаний были диаграммы, таблицы и другие графические элементы, способствующие наглядности представлений. Именно поэтому предлагаемый подход к структурированию связан с возможной визуализацией процесса проектирования.

ОСП позволяет наглядно и компактно отобразить объекты и отношения предметной области на основе использования шести постулатов.

Объектно-структурный подход подразумевает интегрированное использование сформулированных выше постулатов от первой до последней стадий разработки БЗ интеллектуальных и обучающих систем. На основе ОСП предлагается алгоритм *объектно-структурного анализа* (ОСА) предметной области, позволяющего оптимизировать и упорядочить достаточно размытые процедуры структурирования знаний.

Стратификация знаний

Основы ОСА были предложены автором еще в работах [Гаврилова, 1989; Гаврилова, Красовская, 1990], и успешно применялись при разработке ЭС МИКРОЛЮШЕР [Гаврилова, Тишкин, Золотарев, 1989] и АВЭКС [Гаврилова, Минкова, Карапетян, 1992].

ОСА подразумевает дезагрегацию ПО, как правило, на восемь страт или слоев (табл. 2.1 и 2.2).

Таблица 2.1. Стратификация знаний предметной области

Страта	Вид знаний страты	Уровни страты
s_2	ЗАЧЕМ-знания	Стратегический анализ: назначение и функции системы
s_2	КТО-знания	Организационный анализ: коллектив разработчиков системы
s_3	ЧТО-знания	Концептуальный анализ: основные концепты, понятийная структура
s_4	КАК-знания	Функциональный анализ: гипотезы и модели принятия решения
s_5	ГДЕ-знания	Пространственный анализ: окружение, оборудование, коммуникации
s_6	КОГДА-знания	Временной анализ: временные параметры и ограничения
s_7	ПОЧЕМУ-знания	Каузальный или причинно-следственный анализ: формирование подсистемы объяснений
s_8	СКОЛЬКО-знания	Экономический анализ: ресурсы, затраты, прибыль, окупаемость

Объектно-структурный анализ подразумевает разработку и использование матрицы ОСА (табл. 2.2), которая позволяет всю собранную информацию дезагрегировать последовательно по слоям-стратам (вертикальный анализ), а затем по уровням — от уровня проблемы до уровня

подзадачи (горизонтальный анализ). Или наоборот — сначала по уровням, а потом по стратам.

Таблица 2.2. Матрица объектно-структурного анализа

Уровни страты	Уровень области	Уровень проблем	Уровень	Уровень подзадач	...	u_n
Стратегический анализ s_1	E_{11}	E_{12}	E_{13}	E_{14}		E_{1n}
Организационный анализ	E_{21}					
Концептуальный анализ s_3	E_{31}					
Функциональный анализ s_4	E_{41}					
Пространственный анализ	E_{51}					
Временной анализ s_6	E_{61}					
Кавзальный анализ s_7	E_{71}					
Экономический анализ s_8	E_{81}					
.....					E_{ii}	
s_m	E_{m1}					E_{mn}

При необходимости число страт может быть увеличено. В свою очередь знания каждой страты подвергаются дальнейшему ОСА и декомпозируются на составляющие $\|e_{mn}\|$, где m — номер уровня, n — номер страты, а e_{mn} принадлежит множеству K всех концептов (понятий) предметной области.

Алгоритм ОСА (объектно-структурного анализа)

Алгоритм ОСА предназначен для детального практического структурирования знаний предметной области (ПО). В основе ОСА заложен алгоритм заполнения ОСА-матрицы E_{mn} . Алгоритм содержит последовательность аналитических процедур, позволяющих упростить и оптимизировать процесс структурирования. Алгоритм разделяется на две составляющие:

- **A_I:** Глобальный (вертикальный) анализ, включающий разбиение ПО на методологические страты (Что-знания, Как-знания и т. д.) на уровне всей ПО. В результате заполняется первый столбец матрицы.
- **A_II:** Анализ страт (горизонтальный), включающий построение многоуровневых структур по отдельным стратам. Число уровней n определяется особенностями стратифицированных знаний ПО и может существенно отличаться для разных страт. С точки зрения методологии $n < 3$

свидетельствует о слабой проработке ПО.

Первый уровень соответствует уровню всей ПО (предметной области). Второй — уровню проблемы, выделенной для решения. Третий — уровню конкретной решаемой задачи. Дальнейшие соответствуют подзадачам, если имеет смысл их выделять.

При этом возможно как последовательное применение восходящей (bottom-up) и нисходящей концепции (top-down), так и их одновременное применение.

Глобальный анализ

Технология глобального анализа сводится к разбиению пространства основной задачи структурирования ПО на подзадачи, соответствующие особенностям ПО. Для разработки интеллектуальных систем существует минимальный набор s -страт, обеспечивающий формирование БЗ. Минимальный набор включает три страты:

- s_3 , — формирование концептуальной структуры Sk ;
- s_4 — формирование функциональной структуры Sf ,
- s_5 — формирование подсистемы объяснений S_0 .

Формирование остальных страт позволяет существенно оптимизировать процесс разработки и избежать многих традиционных ошибок проектирования. Страты s_4 и s_5 являются дополнительными и формируются в случаях, когда знания предметной области существенно зависят от временных и пространственных параметров (системы реального времени, планирование действий роботов и т. п.).

Алгоритм $A_{_1}$ глобального анализа может быть кратко сформулирован следующим образом:

- $A_{_1_1}$: Собрать все материалы, полученные по результатам извлечения знаний.
- $A_{_1_2}$: Выбрать набор страт N , подлежащих формированию ($N_{\min} = 3$).
- $A_{_1_3}$: Отобрать всю информацию по первой выбранной

страте ($i = 1$, где i — номер из выбранного набора страт N).

➤ A_{1_4} : Повторить шаг A_{1_3} для $i + 1$ для всех выбранных страт до $i \leq N$.

➤ A_{1_5} : Если часть информации останется неиспользованной, увеличить число страт и повторить для новых страт шаг A_{1_3} ; иначе перейти к последовательной реализации алгоритмов горизонтального анализа страт A_2 .

Анализ страт

Последовательность шагов горизонтального анализа зависит от номера страты, но фактически сводится к реализации дуальной концепции структурирования для решения конкретной подзадачи.

Ниже предлагается алгоритм ОСА для одной из обязательных страт S_3 , (ЧТО-анализ), результатом которого является формирование концептуальной структуры предметной области S_k .

➤ $A_{2_3_1}$: Из группы информации, соответствующей ЧТО-страте, выбрать все значимые понятия и сформулировать соответствующие концепты.

➤ $A_{2_3_2}$: Выявить имеющиеся иерархии и зафиксировать их графически в виде структуры.

➤ $A_{2_3_3}$: Детализировать концепты, пользуясь нисходящей концепцией (top-down).

➤ $A_{2_3_4}$: Образовать метапонятия по восходящей концепции (bottom-up).

➤ $A_{2_3_5}$: Исключить повторы, избыточность и синонимию.

➤ $A_{2_3_6}$: Обсудить понятия, не вошедшие в структуру S_f , с экспертом и перенести их в другие страты или исключить.

➤ $A_{2_3_7}$: Полученный граф или набор графов разделить на уровни и обозначить согласно матрице ОСА.

Контрольные вопросы:

1. Приведите пример дуальной стратегии проектирования
2. Какие существуют подходы к проектированию сложных систем.
3. Опишите алгоритм ОСА

Литература:

1. А. П. Частиков Т. А. Гаврилова Д. Л.Белов. РАЗРАБОТКА ЭКСПЕРТНЫХ СИСТЕМ. СРЕДА CLIPS. СПб: БХВ-Петербург, 2003

Ключевые слова:

Структурирование знаний, подходы к проектированию сложных систем, стратификация знаний.