



# ИНТЕГРИРОВАННАЯ АСУ: МЕТОДОЛОГИЯ СОЗДАНИЯ СИСТЕМ, ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, СРЕДСТВ И КОМПОНЕНТОВ

УДК 004.38

А.А. Куприянов, А.С. Мельниченко

## МОДЕЛИ СТРУКТУРИЗАЦИИ И ФОРМАЛИЗАЦИИ ОНТОЛОГИИ ПРЕДМЕТНОЙ ОБЛАСТИ НА СТАДИЯХ ПРОЕКТИРОВАНИЯ АВТОМАТИЗИРОВАННЫХ СИСТЕМ

**Куприянов Анатолий Александрович**, кандидат технических наук, доцент, окончил радиотехнический факультет Ульяновского политехнического института. Ведущий научный сотрудник ФНПЦ ОАО «НПО «Марс». Специализируется в области методологии проектирования и разработки распределенных вычислительных систем. Имеет публикации в области проектирования и разработки локальных и корпоративных сетей, комплексов средств автоматизации и автоматизированных систем управления специального и общего назначения. [e-mail: aakupr1828@rambler.ru].

**Мельниченко Анатолий Степанович**, окончил факультет автоматики и вычислительной техники Московского института инженеров железнодорожного транспорта. Старший преподаватель кафедры телекоммуникационных технологий и сетей Ульяновского государственного университета. Специализируется в области моделирования процессов технологической подготовки разработки программных продуктов и экспертных систем. Имеет научные работы и статьи в области программной инженерии. [e-mail: masulgu@yandex.ru].

### Аннотация

В статье обсуждается проблема сокращения ошибок при создании онтологии на ранних стадиях проектирования автоматизированных систем (АС), в том числе при определении классов и иерархии классов. Иерархия зависит от возможных способов определения уровня детализации, необходимого для приложений; личных предпочтений аналитиков и иногда от требований по совместимости с другими моделями (словесного описания предметной области (ПрО), инфологических моделей, фреймов, семантических сетей). Рассматриваются несколько руководящих принципов, которые нужно учитывать при разработке и оптимизации онтологии. Акцент делается на следующей особенности структуризации и формализации онтологии предметной области на стадиях проектирования автоматизированных систем: проектирование онтологии — это творческий процесс, который должен сочетаться с объектно-ориентированными методами, способами и подходами структуризации и формализации предметной области.

Ключевые слова: автоматизированная система, онтология, иерархия, страта, предметная область.

### Abstract

The article discusses an issue of error decrease during ontology creation at early stages of computer-aided system design including definition of classes and class hierarchy. Hierarchy depends on possible ways to define a level of detail which is necessary for application, on personal preferences of analysts and sometimes on requirements for compatibility with other models (verbal description of application

domain, infological models, frames, semantic networks). It also deals with multiple guidelines to be taken into account during development and optimization of ontology. The following features of structuring and formalization of application-domain ontology at stages of design of computer-aided systems, are stressed: design of ontology is a constructive process which is to combine with object-oriented methods, ways and approaches of structuring and formalization of application domain.

Key words: computer-aided system, ontology, hierarchy, strata, application domain.

### Вводная часть

Такие особенности современных автоматизированных систем, как виртуальность, многоагентность и интеллектуальность рассматриваются как конкурентное их преимущество [1]. Одним из перспективных направлений в области интеллектуализации АС и управления знаниями<sup>1</sup>, которое дает возможность использования накопленных знаний для их компьютерной обработки, являются онтологии [2, 3, 4].

Онтология — это целостная структурная спецификация некоторой предметной области АС, ее формализованное представление, которое включает словарь (или имена) указателей на термины предметной области и логические выражения, описывающие как они соотносятся друг с другом, в том числе:

- каталоги на основе ID;
- словари терминов;
- глоссарии;
- неформальные таксономии;
- формальные таксономии;
- формальные экземпляры;
- свойства на основе фреймов;
- ограничения на основе термов;
- дизъюнктивные классы, обратные свойства;
- произвольные логические ограничения.

Рассмотрим некоторые понятия подробнее:

• **Каталоги на основе ID** — это конечный список терминов (простейшим примером является каталог на основе идентификаторов).

• Другой спецификацией онтологии может быть **глоссарий**, представляющий собой список терминов с их значениями.

• **Тезаурусы** несут дополнительную семантику, определяя связи между терминами.

• Наиболее полно при проектировании систем применяются **формальные таксономии**. Эта разновидность онтологии включает точное определение отношения, обозначаемого как «isA», и строгую иерархию классов, необходимую при использовании наследования для процедуры логического вывода. Следующее понятие — на-

личие в онтологической системе формального отношения, обозначаемого как «isInstanceOf». Далее среди структурных элементов появляются **слоты**. Здесь классы (иногда их называют *фреймами*) могут иметь информацию о свойствах (слотах).

• Возникает необходимость описывать более сложные факты, выразительные средства онтологии (и ее структуры). Это положение требует ее структуризации с последующей формализацией, где некоторые описания онтологии позволяют делать произвольные логические утверждения о концептах — **аксиомы** (существуют языки описания онтологии, такие как CycL и Ontolingua, которые позволяют фиксировать утверждения на языке логики предикатов первого порядка FOL) и вводить **функции**.

Таким образом, основными компонентами онтологии могут являться:

- классы (или понятия);
- отношения (или свойства, атрибуты);
- функции;
- аксиомы;
- экземпляры (или индивиды).

Совместное использование заинтересованными лицами или программными агентами перечисленных основных компонентов предметной области АС является одной из наиболее общих целей разработки онтологии.

В проектировании онтологии условно можно выделить два направления, до некоторого времени развивавшихся отдельно. Первое связано с представлением онтологии как формальной системы, основанной на математически точных аксиомах. Второе направление развивается в рамках компьютерной лингвистики и когнитивной науки. В настоящее время идет поиск связей, позволяющих комбинировать соответствующие методы. Поэтому иногда бывает сложно отделить лексические онтологии с элементами формальных аксиоматик от логических систем с включениями лингвистических знаний.

Онтологический инжиниринг строится на основе следующих принципов [2, 3]:

- **обозначение целей и области применения создаваемой онтологии;**
- **формализация**, т. е. описание объективных элементов действительности в единых, строго определенных образцах (терминах, моделях и др.);
- **использование ограниченного количества базовых терминов (сущностей)**, на основе которых конструируются все остальные понятия;

<sup>1</sup> **Управление знаниями (Knowledge Management)** — совокупность процессов и технологий, предназначенных для выявления, создания, распространения, обработки, хранения и предоставления для использования знаний.

**Задачи и цели системы управления знаниями (СУЗ).** Задача СУЗ — накапливать не разрозненную информацию, а структурированные, формализованные знания — закономерности и принципы, позволяющие решать реальные задачи в различных организациях, в том числе военных, для принятия решений. Основная цель СУЗ — сделать знания доступными и повторно используемыми на уровне всей организации.

- внутренняя полнота и логическая непротиворечивость;
- глубокий структурный анализ предметной области;
- совместное использование пользователями или программными агентами общего понимания структуры информации;
- обеспечение возможности использования знаний ПрО;
- создание явных допущений в ПрО, лежащих в основе реализации онтологии;
- отделение знаний ПрО от оперативных знаний — это еще один вариант общего применения онтологии;
- анализ знаний в ПрО.

Для любой предметной области может существовать достаточно большое количество онтологий. Ведь каждая новая онтология — это один из способов структурирования концепций и отношений ПрО. Однако существуют несколько простых принципов, которые могут помочь при принятии решений о том, как создавать те или иные онтологии:

- не может быть только одного способа описания модели предметной области — всегда есть жизнеспособная альтернатива;
- процесс разработки обязательно должен быть итеративным;
- концепции в онтологии должны быть максимально близки к объектам (логическим или физическим) и отношениям между ними в интересующей предметной области.

Методы и способы создания онтологии хорошо изучены [2, 3, 4]. Однако существует большое количество проблем, связанных с обеспечением логической правильности онтологии, в том числе: создание иерархии классов, обеспечение транзитивности иерархических отношений, развитие иерархии классов, различие между классами и их именами, избежание циклов классов, анализ узлов-братьев в иерархии классов, множественное наследование, ограничение масштаба и т. д.

Рассмотрим их подробнее:

1. Распространенная ошибка при моделировании — это включение в иерархию варианта одного и того же понятия как в единственном, так и во множественном числе, сделав первое подклассом второго. Как только иерархия представляет собой отношение «kind-of», то ошибка при моделировании становится очевидной.

2. Соблюдение транзитивности иерархических отношений, т. е. отношение подкласса транзитивно: если В — это подкласс А, а С — подкласс В, то С — подкласс А.

3. Поддержание последовательной иерархии классов может вызывать сложности по мере того, как развиваются предметные области.

4. Важно различать классы и их имена: классы представляют понятия предметной области, а не слова, которые обозначают эти понятия.

В действительности нужно все время соблюдать правило:

- синонимы одного и того же понятия не представляют различные классы;
- синонимы — всего лишь разные имена понятия или термина. Многие системы позволяют ассоциировать с классом список синонимов, переводов или имен представления. Если система не позволяет осуществлять такие ассоциации, то синонимы всегда можно перечислить в документации к классу.

5. Следует избегать циклов в иерархии классов (в иерархии есть цикл, когда у некоторого класса А есть подкласс В и в то же время В — это подкласс А).

6. Узлы-братья в иерархии — это классы, которые являются прямыми подклассами одного и того же класса. Все узлы-братья в иерархии (кроме тех, что находятся в корне) должны располагаться на одном уровне обобщения.

7. Большинство систем представления знаний позволяет осуществлять множественное наследование в иерархии классов: класс может быть подклассом нескольких классов.

8. Одно из самых сложных решений, которое нужно принять во время моделирования, — это определить, когда ввести новый класс или когда сформулировать различие с помощью разных значений свойств. Найти подходящий баланс нелегко.

9. При моделировании предметной области нужно решать, моделировать ли определенное разграничение как значение свойства или как набор классов, что зависит от масштаба предметной области и от решаемой задачи.

10. Онтология не должна содержать всю возможную информацию о предметной области: не нужно конкретизировать или обобщать больше, чем нужно для приложения (не более 1 дополнительного уровня в каждую сторону).

11. Многие системы позволяют явным образом задать, что несколько классов являются дизъюнктивными. Классы дизъюнктивные, если у них не может быть общих экземпляров.

Таким образом, при разработке онтологии самое узкое место — **не программный аспект, а задача извлечения, формулирования, структурирования и представления информации предметной области, т. е. данных и знаний, с соблюдением логических правил и условий (правильно построенных формул) с целью устранения вышеперечисленных в пунктах 1-11 недостатков.** Поэтому акцент на эти особенности структуризации и формализации онтологии предметной области делается на стадиях проектирования автоматизированных систем: проектирование онтологии — это творческий процесс, который должен сочетаться с объектно-ориентированными методами, способами и подходами [2, 6].

Основные этапы методики структуризации и формализации онтологии

При разработке систем управления знаниями на основе онтологии, как правило, выделяют следующие этапы [4]:

- **накопление** — стихийное и бессистемное накопление информации в организации;
- **извлечение** — процесс, идентичный традиционному извлечению знаний для экспертной системы (один из наиболее сложных и трудоемких этапов, от его успешности зависит дальнейшая жизнеспособность системы);
- **структурирование** — на этом этапе должны быть выделены основные понятия, выработана структура представления информации, обладающая максимальной наглядностью, простотой изменения и дополнения;
- **формализация** — представление структурированной информации в форматах машинной обработки, то есть на языках описания данных и знаний;
- **обслуживание** — корректировка формализованных данных и знаний (добавление, обновление), удаление устаревшей информации, фильтрация данных и знаний для поиска информации, необходимой пользователям.

Структурным анализом принято называть метод исследования системы, который начинается с ее общего обзора. Результаты исследования затем детализируются, приобретая иерархическую структуру со все большим числом уровней. Для таких методов характерно: разбиение на уровни абстракции с ограниченным числом элементов (от 3 до 7 так называемых страт [4]); использование строгих формальных правил записи; последовательное приближение к результату.

Существуют различные методологии структурного моделирования предметной области, среди которых следует выделить функционально-ориентированные и объектно-ориентированные методологии. В данной работе последовательно применяется объектно-ориентированная методология. В таблице 1 представлена последовательность этапов структуризации и формализации онтологии на основе объектно-ориентированной методологии, рассмотренной в [4, 6].

**Разработка общей матричной модели стратификации предметной области (онтологии)**

В основе общей матричной модели стратификации лежит матричная модель Захмана (рис. 1), усовершенствованная в работе [4], приближая ее к моделям стратификации онтологии.

Объектно-структурный анализ (ОСА) подразумевает разработку и использование матрицы объектно-структурного анализа (таблица 2), которая позволяет всю собранную информацию дезагрегировать последовательно по слоям — стратам (вертикальный анализ), а затем по уров-

ням — от уровня проблемы до уровня понятий (горизонтальный анализ). Или наоборот — сначала по уровням, а потом по стратам.

**Разработка детализированной матричной модели предметной области (онтологии)**

В основе объектно-структурного анализа заложен алгоритм заполнения матрицы  $E_{mn}$  (таблица 3). Алгоритм ОСА содержит последовательность процедур, позволяющих упростить и оптимизировать процесс структурирования. Алгоритм разделяется на две составляющие:

- **глобальный (вертикальный) анализ**, включающий разбиение предметной области на методологические страты на уровне всей ПрО. В результате заполняется первый столбец матрицы;
- **анализ страт (горизонтальный)**, включающий построение многоуровневых структур по отдельным стратам. Число уровней  $n$  определяется особенностями стратифицированных знаний ПрО и может существенно отличаться для разных страт.

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

Таблица 1  
Последовательность этапов структуризации и формализации онтологии

Последовательность этапов	Номер этапа	Содержание этапа
Этапы структуризации	1	Разработка описаний объектов предметной области на русском языке в виде словаря и глоссария.
	2	Разработка общей матричной модели стратификации предметной области (онтологии).
	3	Разработка детализированной матричной модели предметной области (онтологии).
Этапы формализации	4	Разработка логико-алгебраического представления компонент онтологии.
	5	Разработка детализированной матричной модели онтологии.
	6	Разработка оптимизированной статической модели предметной области (онтологии).

множеству  $K$  всех концептов (понятий) предметной области.

#### Разработка логической модели онтологии

Набор выражений исчисления предикатов может быть использован для определения отношений между объектами предметной области в следующей последовательности [6]:

1. Формирование на основе семантических моделей и правильно построенных формул логико-алгебраического представления предметной области.

2. Минимизация логико-алгебраических представлений предметной области.

3. Преобразование минимизированного логико-алгебраического представления объектов предметной области в онтологию.

#### Разработка модели оптимизации онтологии

Решение задачи оптимизации является одной из функций математической логики. В ее основе лежит использование математической модели и логического представления статической модели для выбора проектного решения.

Решение подобных задач оптимизации в общем случае может быть выполнено на основе аналитического и аналитико-эвристического подходов.

При аналитическом подходе выбор аргументированной аксиоматики и вся процедура поиска оптимального решения происходит по следующей схеме:

$$S \rightarrow A \rightarrow \text{opt } y \rightarrow y^0 \rightarrow x^0;$$

где  $S$  – некоторая ситуация принятия решения,

$A$  – аксиоматика компромисса;

$x^0$  – оптимальное решение.

При аналитико-эвристическом подходе переход от  $S$  к  $A$  осуществляет человек, анализируя ситуацию и из предлагаемого набора аксиоматик выбирая наиболее подходящую, используя базу знаний экспертной системы, которая в этом случае является инструментом принятия решения.

Для использования первого подхода пока нет теоретической базы, поэтому целесообразно использовать второй подход (о математической модели оптимизации см. работу [6]).

#### Разработка статической модели предметной области

Статическая модель отражает основные понятия и элементы, характерные для данной предметной области, с формализованным описанием взаимосвязей между ними, а также методов их задания и обработки. Статическая модель описывается средствами языка UML [4, 5].

UML – стандартный язык для написания моделей анализа, проектирования и реализации объектно-ориентированных программных систем. Словарь UML образует три разновидности строительных блоков: предметы, отношения, диаграммы. В UML имеются четыре разновидности предметов:

- структурные предметы;
- предметы поведения;
- группирующие предметы;
- поясняющие предметы.

Эти предметы являются базовыми объектно-ориентированными блоками для разработки моделей.

Представление стратифицированных знаний о предметной области на языке UML приведено в таблице 3.

	Данные «ЧТО»	Функции «КАК»	Дислокация, сеть «ГДЕ»	Люди «КТО»	Время «КОГДА»	Мотивация «ПОЧЕМУ»	
Планировщик	Список важных понятий и объектов	Список основных бизнес-процессов	Территориальное расположение	Ключевые организации	Важнейшие события	Бизнес-цели и стратегии	Сфера действия (контекст)
Владелец, менеджер	Концептуальная модель данных	Модель бизнес-процессов	Схема логистики	Модель потока работ (workflow)	Мастер-план реализации	Бизнес-план	Модель предприятия
Конструктор, архитектор	Логические модели данных	Архитектура приложений	Модель распределенной архитектуры	Архитектура интерфейса пользователя	Структура процессов	Роли и модели бизнес-правил	Модель системы
Проектировщик	Физическая модель данных	Системный проект	Техническая архитектура	Архитектура презентаций	Структуры управления	Описания бизнес-правил	Технологическая (физическая) модель
Разработчик	Описание структуры данных	Программный код	Сетевая архитектура	Архитектура безопасности	Определение временных привязок	Реализация бизнес-логики	Детали реализации
	Данные	Работающие программы	Сеть	Реальные люди, организации	Бизнес-события	Работающие бизнес-стратегии	Работающее предприятие
	Данные	Функции, процессы	Сеть, расположение систем	Люди, организации	Время, расписание	Мотивация	

Рис. 1. Структурная модель Захмана

Таблица 2  
Стратификация знаний предметной области

Страта	Уровни страты
s_1	Стратегический анализ: назначение и функции системы.
s_2	Организационный анализ: коллектив разработчиков системы.
s_3	Концептуальный анализ: основные концепты, понятийная структура.
s_4	Функциональный анализ: модели принятия решения.
s_5	Пространственный анализ: окружение, оборудование, коммуникации.
s_6	Временной анализ: временные параметры и ограничения.
s_7	Каузальный или причинно-следственный анализ.
s_8	Экономический анализ: ресурсы, затраты.

Таблица 3  
Представление стратифицированных знаний о предметной области на языке UML

Страты	Сущности языка UML		
	Предметы	Отношения	Диаграммы
s_1	Структурные предметы, группирующие предметы.	Зависимость, ассоциация, обобщение, реализация.	Диаграммы прецедентов.
s_2	Структурные предметы, группирующие предметы.	Зависимость, ассоциация, обобщение, реализация.	Диаграммы классов и объектов.
s_3	Структурные предметы.	Зависимость, ассоциация, обобщение, реализация.	Диаграммы прецедентов (список ролей и актеров).
s_4	Структурные предметы.	Зависимость, ассоциация, обобщение, реализация.	Диаграммы размещения, компонентная модель.
s_5	Поясняющие предметы.	Зависимость, ассоциация, обобщение, реализация.	Диаграммы последовательности.
s_6	Предметы поведения.	Зависимость, ассоциация, обобщение, реализация.	Диаграммы прецедентов.
s_7	Предметы поведения, поясняющие предметы.	Зависимость, ассоциация, обобщение, реализация.	Диаграммы прецедентов.

## ЗАКЛЮЧЕНИЕ

Перечисленный комплекс моделей позволяет наглядно и компактно отображать объекты и отношения предметной области в процессе проектирования онтологии АС. Особенность предлагаемой методики моделирования состоит в том, что:

1. Разработка ряда моделей выполняется на основе объектно-ориентированной методологии моделирования на языке UML. Объектно-структурный подход подразумевает интегрированное использование сформулированных выше методов и средств, позволяющих оптимизировать и упорядочить процедуры структурирования предметной области.

2. Для принятия решения по выбору оптимальной модели предметной области предварительно разрабатывается модель на языке исчисления предикатов в виде правильно построенных формул.

Применение предложенной методики моделирования, структуризации, формализации позволит в конечном итоге сократить время и стоимость создания онтологии и избежать многочисленных логических ошибок на стадиях проектирования автоматизированных систем.

## СПИСОК ЛИТЕРАТУРЫ

1. Куприянов А. А. Подход к созданию виртуальной организации проектирования и изготовления программных изделий ИАСУ / А. А. Куприянов, А. С. Мельниченко, А. Ю. Крайнов // Автоматизация процессов управления. — 2009. — № 3 (17). — С. 33–43.
2. Муромцев Д. И. Онтологический инжиниринг знаний в системе *Protégé* / Д. И. Муромцев. — СПб. : СПб ГУ ИТМО, 2007. — 62 с.
3. Гладун А. Я. Онтологии в корпоративных системах. Часть II / А. Я. Гладун, Ю. В. Рогушина // Корпоративные системы. — 2006. — № 1. — С. 20–26.
4. Частиков А. П. Разработка экспертных систем. Среда CLIPS / А. П. Частиков, Т. А. Гаврилова, Д. Л. Белов. — СПб. : БХВ-Петербург, 2003. — 608 с.
5. Орлов С. А. Технология разработки программного обеспечения / С. А. Орлов. — СПб. : Питер, 2002. — 464 с.
6. Мельниченко А. С. Модели оптимизации процессов создания программных продуктов / А. С. Мельниченко // Ученые записки Ульяновского государственного университета. Сер. Информационные технологии. — 2005. — Вып. 2. — С. 102–108.