

УДК 004.318

А.В. Палагин, Н.Г. Петренко

Системно-онтологический анализ предметной области

Введение. Согласно [1] под *анализом* понимается вид исследования, при котором реальный или мыслимый объект расчленяется на составляющие части (элементы) и исследуются эти элементы и связи между ними. Анализ предметной области (ПдО) представляет особый вид научной деятельности, в результате которой строится интерпретационная модель предметных знаний (в широком смысле). В процессе анализа последние делятся на инвариантные и прагматические знания, концептуальные составляющие которых представляют онтологические знания ПдО.

Некоторые идеи по разработке методологии проектирования онтологии ПдО берут своё начало в литературе по объектно-ориентированному подходу (ООП), возникшему как технология программирования больших программных продуктов [2]. Однако разработка онтологий как иерархической структуры понятий (концептов) отличается от проектирования объектов как классов и отношений в объектно-ориентированном программировании. Последний сосредотачивается главным образом на методах описания классов – программист принимает проектные решения, основанные на *операторных* свойствах класса, тогда как разработчик онтологии принимает эти решения, основываясь на *структурных* свойствах класса. В результате структура понятий и отношения между понятиями в онтологии отличаются от структуры классов объектов подобной ПдО в объектно-ориентированной программе [3]. Кроме того, при разработке онтологии внутреннее содержимое понятия эксплицируется всегда, в то время как в объектно-ориентированном программировании зачастую применяется метод инкапсуляции как способ ограничения доступа к внутреннему содержимому объекта.

Системный подход к познанию ориентирует аналитика на рассмотрение любой ПдО с позиций закономерностей системного целого и взаимодействия составляющих его частей. Системность знаний исходит из многоуровневой иерархической организации любой сущности, то есть все объекты, процессы и явления можно рассматривать как множество более

мелких подмножеств (признаков, деталей) и, наоборот, любые объекты можно (и нужно) рассматривать как элементы более высоких классов обобщений.

Начало 90-х годов прошлого столетия считается зарождением парадигмы компьютерных онтологий. Она была сформулирована как попытка сгладить (и по возможности устранить) всё чаще проявляющиеся разного рода противоречия при функционировании и внедрении интеллектуальных систем с использованием баз знаний предметных областей. Ярким представителем таких систем на то время были экспертные системы (ЭС). Они успешно и эффективно функционировали в пределах одного коллектива и на уровне коммерческих образцов. У разработчиков постоянно возникал вопрос: “Как обеспечить продвижение этих образцов к конечному пользователю?”. Были предложены разработки оболочек - “пустых” ЭС и ряд других новшеств. Но они не имели решающего значения. Напрашивался вывод, что для эффективного функционирования ЭС на самом важном этапе “жизненного цикла” – функционирования у конечного пользователя при решении реальных задач – необходимо к каждой ЭС “приложить” эксперта в соответствующей ПдО. Необходимость присутствия эксперта объяснялась, в частности, быстротечностью изменения знаний во многих предметных областях и соответственно необходимостью обновления базы знаний ЭС в “реальном времени”. В тех же ПдО, где знания обладали относительной инвариантностью, ЭС продолжали эффективно функционировать.

Сказанное выше и ряд других факторов “подтолкнули” учёных к разработке парадигмы *компьютерных онтологий*, основные принципы которой были сформулированы в [4].

1. Доходчивость, ясность (Clarity). Термины (и понятия) онтологии должны отражать реальную действительность. Их символные обозначения (знаки) должны формироваться на основе общепринятых правил в семиотике и должны выражать общепринятые смыслы реальных объектов. В свою очередь, эти смыслы извлекаются из общепринятых определений терминов (понятий), зафиксированных в толковых словарях, различных глоссариях ПдО. Суждения, входящие в определения, формализуются на основе формального общепринятого аппарата в виде тождественно истинных логических аксиом.

2. **Обоснованность, связность (Coherency).** Формирование начального набора понятий онтологии и их добавление должно быть обоснованным, определяемым, в первую очередь, требованиями предполагаемой совокупности решаемых задач. Логические аксиомы начального набора понятий должны быть непротиворечивыми. Для этого должен быть предусмотрен механизм логического вывода, который, в том числе, проверяет на непротиворечивость добавляемые аксиомы и выводимые в онтологии утверждения.

3. **Расширяемость (Extendibility).** Ядром онтологии являются первоначально введенные (спроектированные) понятия и описывающие их аксиомы. В онтологии должен быть предусмотрен механизм расширения (ограничения) совместно используемых словарей понятий без нарушения целостности системы.

4. **Минимальное влияние кодирования (Minimal encoding bias).** В онтологической системе (ОнС) должен быть реализован принцип совместного использования онтологий, который предполагает: спецификацию онтологии на уровне представления, а не символического кодирования; запись такой спецификации на общепринятом и платформонезависимом языке описания онтологий можно передать для использования любому программному агенту.

5. **Минимальные онтологические обязательства (Minimal ontological commitment).** Этот принцип перекликается с принципами обоснованности и расширяемости/ограничения. Важно, чтобы множество понятий онтологии отображало концептуальную структуру ПдО, относительно стабильную на протяжении “жизненного цикла” ОнС. А последняя предоставляла возможность расширения или специализации отдельных ветвей онтологического графа (онтографа, ОГ). Отделение концептуальных знаний от знаний, выраженных фактами, является стратегией построения ОнС, а точнее – онтологических баз знаний.

Постановка задачи. Известны общие совокупности методов, принципов, процедур и атрибутов системного анализа как научного познания в любой ПдО [5, 6 и др.]. При конкретизации ПдО конкретизируются и средства системного анализа. Парадигма компьютерных онтологий, развиваемая во взаимодействии со средствами и методами системного анализа, поло-

жила начало развития новой ветви средств и методов системного анализа ПдО – системно-онтологического анализа (или подхода).

Центральной идеей системно-онтологического подхода (СОП) является разработка онтологических средств поддержки решения прикладных задач – полифункциональной онтологической системы¹. Такая система (точнее, её концептуальная часть) описывается кортежем (1) и представляет онтологию ПдО, состоящую из онтологии объектов и онтологии процессов, и онтологию задач [7, 8].

(1)

На рис.1 представлена схема компонентов проблемного пространства (ПрП) и соответствующие ему онтологии. ПрП – это модель всех таких аспектов или компонент ПдО, с которыми связаны (опосредственно или непосредственно) знания, требующиеся при решении различных задач в этой ПдО. Всякое ПрП состоит из двух блоков: инвариантной (относительно неизменной) части и множества изменяемых частей, соответствующих отдельным задачам. В составе инвариантной части, например в методологии SMEE (Structured Methodology for Elicitation of Expertise), выделяют семь типов компонент: объекты, инструменты, операторы, операции, конечные продукты, побочные продукты и ограничения [9]. Эти типы компонент – суть понятия, которые хорошо группируются в онтологии объектов и процессов, представленных на рис.1. На нём приняты следующие обозначения:

- O^O – онтология множества объектов (понятий, концептов) ПдО, которая рассматривается как иерархическая структура классов, подклассов и элементов классов;
- O^P – онтология множества процессов ПдО, которая рассматривается как иерархическая структура процессов, подпроцессов, действий и операций;
- O^3 – онтология совокупности задач, которые могут быть поставлены и решены в ПдО. Рассматривается как иерархическая структура задач, подзадач, процедур и операторов.

¹ В данной работе не рассматриваются такие компоненты онтологической системы как машина вывода и метаонтология.

Рис.1. Схема компонентов проблемной области

Итак, *главной задачей исследования* является разработка формализованной методики.

Общий подход к проектированию. Известна методология структурного анализа и проектирования (SADT) сложных систем в произвольной предметной области [11]. Эта методология породила семейство методик (и соответствующих стандартов) IDEF (Integrated DEFinition), ориентированных на разработку моделей ПдО и акцентирующих внимание на каком-то конкретном аспекте проектирования [12, 13]. В частности:

- методика IDEF0 рекомендована для содержательного анализа и функционального проектирования сложных систем управления, в том числе и программного обеспечения. Однако в последнем случае трудно определить её преимущества по сравнению с методологией ООП. Описание объектов и процессов в методике выполняется в виде иерархической совокупности диаграмм с лаконичным описанием функций. Блоки на диаграммах выражают функции, поэтому их названия – глаголы или отглагольные существительные;
- методика IDEF1x предназначена для информационного моделирования, основана на концепции “сущность-связь”. Обычно отправным пунктом для разработки информационной модели является IDEF0-модель;
- методика IDEF3 описывает поведенческие аспекты конкретных приложений, рассматривает последовательность выполнения и причинно-следственные связи между ситуациями и событиями для структурного представления знаний о ПдО. Если IDEF0 связана с функциональными аспектами и отвечает на вопросы “Что делает система?”, то в IDEF3 детализируются IDEF0-функции. Эта модель отвечает на вопросы “Как система это делает?”;
- методика IDEF5 предназначена для онтологического анализа ПдО, анализа основных терминов и понятий, используемых для описания объектов и процессов, границ использования, а также взаимосвязей между ними. Служит для эффективного исследования и документирования: словаря терминов, используемых при описании характеристик объектов и процессов, имеющих отношение к рассматриваемой ПдО, точных и однозначных определений всех терминов и классификации логических взаимосвязей между ними.

Для каждой из методик семейства IDEF разработаны этапы и стадии построения модели ПдО, языки и диаграммы представления результатов. Анализ получаемых в результате применения методик IDEF моделей ПдО и их описаний с точки зрения формализации и компьютерной обработки показал, что, по сути, они (функциональные модели и описания) представляют собой содержательное техническое задание на проектирование ОнС, включающей онтологию объектов (сущностей), онтологию процессов и онтологию задач. Описание же самой ОнС имеет несколько иные цели. Оно ориентировано на компьютерное представление с помощью стандартизованных языков предметных знаний с целью широкого использования сообществом пользователей.

Итак: 1) методики IDEF и системно-онтологический подход используют одни и те же множества сущностей ПрП, акцентируя внимание на различных совокупностях характеристик и атрибутов; 2) конечные цели этих двух подходов отличаются: для IDEF – функционально-блочная модель, а для СОП – онтолого-содержательная модель.

Выполненный выше анализ позволяет утверждать, что методики IDEF и системно-онтологический подход решают разные задачи, но со значительным “перекрытием” некоторых функций. Весьма очевидно это проявляется для пары “онтология процессов ↔ поведенческая модель ПдО”. На самом деле в ООП (как и в онтологии задач) основными понятийными единицами являются классы объектов (задач) и методы их описания (решения).

Очевидно, этапы проектирования онтологических систем исходят из принятых в методологии SADT фаз проектирования сложных систем: 1) анализ - определение того, что система будет делать; 2) проектирование - определение подсистем и их взаимодействие; 3) реализация – разработка подсистем по отдельности, объединение – соединение подсистем в единое целое; 4) тестирование – проверка работы системы; 5) установка – введение системы в действие; 6) функционирование – использование системы.

1. Предварительный анализ предметной области

Во все методологии включен этап предварительного анализа ПдО или составления содержательного ТЗ на проектирование [11-18 и др.]. Этот этап (как и процесс проектирования базы знаний ПдО в целом) носит сложный аналитический характер и заключается в многократном абстрагировании, в результате которого из всего многообразия сторон и свойств сущностей предметной области выделяются наиболее существенные, релевантные конкретным задачам. Знание предметной области, понимание сути происходящих в ней процессов, законов, правил и ограничений, управляющих её развитием, является необходимым условием успешного решения задач, стоящих перед исследователем. Более того, наличие таких знаний является необходимым условием постановки, формулировки этих задач, без чего невозможно само решение [19]. Онтологические системы призваны сделать знания коллективным достоянием широкого круга лиц, дать мощный инструмент для фиксации, приобретения и обработки знаний, проверки их на непротиворечивость, полноту и т.п. Кроме того, составляется систематизированное представление знаний о ПдО, выявляются источники формирования элементов множеств и процедур, задач, выполняемых в анализируемой ПдО. Составляется и документируется словарь терминов ПдО.

Этап предварительного анализа ПдО включает:

-
-
-

Если проблемное пространство представляет сложную систему, то следует рассмотреть вопрос о предварительном этапе проектирования на основе методик IDEF, которые дополняют описанные выше шаги проектирования. Тем более, что известные инструментальные средства анализа ПдО и построения баз знаний (например, KADS-системы, российская SIMER+MIR) по ряду причин не могут быть использованы.

Как правило, методика сводится к алгоритму, который носит итеративный характер. Для процесса разработки необходимо предусмотреть ряд “контрольных точек” для проверки полученных результатов на соответствие выбранным критериям. Указанные критерии долж-

ны соотноситься с заданными критериями на проектирование базы знаний ПдО, так как создание последней является целью для разработчиков. Оптимальный результат, как правило, зависит от степени проработки предполагаемых приложений и вариантов использования онтологии.

2. Онтология объектов ПдО

Под **онтологией объектов** предметной области понимается кортеж четырёх множеств:

(2)

где

Компьютерная онтология является (формальным) выражением концептуальных знаний о предметной области и по своей значимости сопоставима с базой знаний интеллектуальной информационной системы, а её построение является специфической формой человеческого творчества. Творческий процесс можно представить совокупностью операций-процедур с суждениями, утверждениями, понятиями и отношениями между ними. А последние являются фундаментом, основой для построения составной части научной теории – онтологической базы знаний в заданной предметной области. При этом такие знания описываются в декларативной форме.

Онтология определяет общеупотребительные, семантически значимые “понятийные единицы знаний”, которыми оперируют исследователи и разработчики знаниеориентированных информационных систем. Она отделяет “статические” и “динамические” компоненты знаний ПдО от операциональных знаний. В отличие от знаний, закодированных в алгоритмах, онтология обеспечивает их унифицированное и многократное использование разными исследовательскими группами, на разных компьютерных платформах при решении разных задач.

Построение компонент онтологии

Напомним некоторые известные определения, непосредственно относящиеся к построению множеств концептуальной модели ПдО или её онтологии [20, 21].

Понятие

Все понятия (или концепты) делятся на ряд классов (по семантической зависимости).

1. В зависимости от отображения вида или рода предметов – на *видовые* и *родовые* понятия.
2. В зависимости от отображения части или целого предметов – на *понятия-части* и *понятия-целые*.
3. В зависимости от количества отображаемых предметов – на *единичные* и *общие* понятия.
4. В зависимости от отображения предмета или свойства, абстрагированного от предмета, – на *конкретные* понятия и *абстрактные* понятия.

Онтология ПдО – это концептуальная модель реального мира и её понятия должны отражать эту реальность.

Построение множества X считается наиболее важным моментом при разработке онтологии ПдО. Оно должно быть обязательно не пустым.

Для хорошо проработанных предметных областей за основу множества элементов $\{x_i\}$ может быть взято содержимое подходящих толковых словарей. В противном случае следует составить полный список терминов, в котором указать (причём пересечение объёмов и содержаний понятий в таком предварительном списке не существенно):

- чем является каждый термин – понятием-классом предметов или конкретным понятием;
- указать для каждого термина возможные существенные отношения с другими терминами из списка;
- описать возможные существенные свойства понятий.

Известно, что в любой предметной области существуют термины-синонимы. Для них в онтологии отводится только одно понятие, в аксиомах которого может быть указан синонимический ряд терминов. Другими словами, синонимы одного и того же понятия не представляют различные классы.

Далее следует уточнить и определить окончательный список классов-понятий, имена которых будут входить в разрабатываемую онтологию и являться вершинами онтографа. Также следует принять единые *правила присваивания имён* понятиям и их свойствам.

Затем, возможно, следует повторить некоторые фрагменты процесса анализа ПдО (с привязкой к составленному списку понятий), выполненные на предварительном этапе. Отметим, что в число указанных выше “контрольных точек” (точек входа итерации) должно быть включено завершение разработки любого компонента онтологии.

В результате должен быть получен полный список существенных для заданной ПдО (и предполагаемых приложений) понятий и их машинно-интерпретируемые формулировки.

Построение множества R также основано на результатах предварительного этапа анализа ПдО.

На практике множество R на начальном этапе представляют некоторым обобщённым отношением “*выше – ниже*”. Известно несколько подходов для разработки иерархии классов: процесс *нисходящей* разработки, процесс *восходящей* разработки и *комбинированный* процесс разработки. Последний наиболее часто используется разработчиками, так как он является более естественным, сначала оперирует понятиями среднего уровня, к которым наиболее часто обращаются разработчики. Затем эти понятия обобщаются и ограничиваются.

В заключение данного подэтапа следует соотнести разработанные классы и их иерархии с результатами предварительного анализа ПдО. В частности, уточняются зависимости для конкретных пар (x_i, x_j) . В процессе соотнесения (и построения иерархии) следует учитывать, что [3]:

- прямые подклассы в иерархии должны располагаться на одном уровне обобщения;
- класс может быть подклассом нескольких классов, и тогда он может наследовать свойства от всех этих классов;

- если класс имеет только один прямой подкласс, то, возможно, при моделировании допущена ошибка или онтология неполная;
- если у данного класса есть более дюжины (иногда говорят о числе 7) подклассов, то, возможно, необходимы дополнительные промежуточные классы;
- в онтологии число классов соотносится с числом предполагаемых приложений.

И следует помнить, что не существует единственно правильной иерархии классов.

Описанное построение онтографа является специальным видом классификации понятий ПдО – *онтологической классификацией*.

Построение множеств F и A .

В конечном счёте, независимо от того, какая из этих формулировок будет принята, эффективность разработанной онтологии будет определяться конечными результатами приложений.

В определении концептуальной модели онтологии объектов множество аксиом A состоит из множества определений D_i' и множества ограничений Rs_i' для понятия X_i . Определения записываются в виде тождественно истинных высказываний, которые могут быть взяты, в частности, из толковых словарей ПдО. В них могут быть указаны дополнительные взаимосвязи понятий X_i с понятиями X_j . Во множестве ограничений Rs_i могут быть заданы ограничения на интерпретацию соответствующих понятий X_i .

Кроме того, из полного списка отобранных в онтологию терминов не все представляют понятия. Существуют термины (например, ролевые), которые соответствуют свойствам определённых классов-понятий. Такие свойства следует привязать к описанию самого общего класса, обладающего ими. А подклассы этого класса будут наследовать указанное свойство (конечно, если между ними установлено некоторое отношение частичного порядка).

Свойства понятий имеют определённые значения, такие как *тип значений*, *мощность значений*, *разрешённые значения* (для данного класса) и другие. Например, мощност

ний можно описать: с единичной мощностью, мощностью без ограничений и мощностью с некоторым допустимым интервалом.

На основе построенных множеств кортежа можно синтезировать концептуальную модель ПдО, например, с помощью известного инструментального средства Protégé и получить формальное описание разработанной онтологии на одном из языков описания онтологий, а также графическое представление онтографа.

3. Онтология процессов ПдО

Синонимами онтологий объектов и процессов являются соответственно статическая и динамическая онтологии ПдО. В научно-технической литературе, когда говорят об онтологии ПдО, то подразумевается её статическая составляющая. Именно компоненты последней наиболее разработаны, как в литературе по философии, так и в конкретных описаниях ряда предметных областей. Под поведенческим описанием сущностей-процессов чаще всего понимают разработку определённых бизнес-процессов. Их результатом являются графические диаграммы и естественно-языковые описания процессов. Разработка же базы знаний не является прямой целью указанных методик. Поэтому методики разработки онтологии процессов практически неизвестны. Хотя следует отметить, что в некоторых известных онтологиях верхнего уровня (например, онтология Дж. Совы, SUMO и Mikrokosmos) сущность понятия “Процесс” рассмотрена достаточно детально [22-25].

На рис.2 представлен синтезированный онтограф (на основе анализа указанных онтологий), который представляет схему начального развития понятия “Процесс”, причём только той её части, которая соответствует процессам в научно-технических предметных областях. Ветви онтографа “Социальный процесс”, “Материальный процесс” и им аналогичные не рассматриваются.

Категория *Процесс* рассматривается как *Действительность* и *Событийность*, в отличие от категории *Объект*, характеризующийся как *Действительность* и *Продолжительность* [26]. В первую очередь *Процесс* рассматривается как времязависимая категория и за-

тем подразделяется по видам изменений, наличием начальных и конечных точек и т.д. Далее *Процесс* подразделяется на непрерывный и дискретный. Первый из них характеризуется наличием эксплицитных начальной и конечной точек или как без явного указания этих точек. Второй вид процесса указывает, что изменения происходят дискретными шагами, названными событиями, которые чередуются с периодами покоя, названными состояниями.

Рис.2. Начальное развитие онтологии процессов

Приведенная схема начального развития онтологии процессов не отражает всех характеристик (оснований ветвлений в онтографе) категории “Процесс”, даже для той её части, что представлена на рис.2.

На рис.3. представлена общая схема онтологии процесса ПдО, в которой категория “Процесс” представлена онтографом с p уровнями и n_p подпроцессами (ПП) на каждом уровне. Предпоследний ($p-1$) уровень представлен множествами действий, на которые разбивается каждый ПП предыдущего уровня. В свою очередь, каждое действие на последнем (p -ом) уровне разбивается на последовательность операций $O_{i_j}^p$.

Рис.3. Общая схема онтологии процесса

Связи между подпроцессами для соседних уровней соответствуют отношениям “Целое-часть”, а внутри каждого уровня – некоторой смешанной формой организации соединений. На рис.3 показан частный случай такой организации – параллельный. Дальнейшее развитие (конкретизация) онтологии процессов возможно, когда задана конкретная предметная область и соответствующее проблемное пространство, а в более узком смысле – конкретные признаки ветвлений (условия инициирования ПП, условия окончания ПП и ограничения) в онтографе.

Объём динамических знаний W в предметной области можно оценить через характеристики (параметры) их формально-онтологических представлений. В частности, при представлении онтографом (без учёта сложности функций интерпретации подпроцессов) величи-

на W может характеризоваться числом вершин ОГ. Для ОГ, представленного на рис.3, это число может быть выражено формулой

(3)

где –

3. *Онтология задач ПрП*

В текстовых описаниях (спецификациях) целевых задач выделяются набор объектов и набор процессов (методов), необходимых и достаточных для выполнения конкретных целевых задач. Можно выделить некоторое унифицированное (пополняемое) множество базовых задач (типовых фрагментов задач), на основе которых с помощью определённых логических последовательностей конструировать более сложные задачи. Таким образом, решение прикладной задачи включает, в том числе, взаимодействие трёх онтологий: онтологии объектов, онтологии процессов и онтологии базовых задач [7, 8].

Онтологические знания, описывающие некоторое ПрП, можно разделить на следующие компонентные знания: типы входных и выходных данных, инструменты, операторы (человек или компьютерная программа) и операции (действия оператора или *Решателя задач*) [9].

Для реализации подхода необходимо разработать унифицированный язык представления онтологических знаний и инструментальную среду как набор специализированных и универсальных базовых операций, управляющих процессом решения. Необходимо также разработать *Формирователь задач*, который осуществляет выбор средств и методов формирования структуры задачи на основе базовых операций.

Схема модели онтологии задач описывается кортежем

(4)

где

На рис.4 представлена схема онтологии задач.

Одно из преимуществ онтологического подхода, в частности, иерархического представления состоит в том, что сложная задача большой размерности разбивается на последовательно решаемые группы задач малой размерности. Онтология задач в качестве понятий содержит типы решаемых задач, а отношения этой онтологии, как правило, специфицируют декомпозицию задач на подзадачи.

На рис.5 представлен алгоритм методики проектирования онтологии ПдО и онтологии задач. Предполагается, что множество функций интерпретации в моделях онтологий тождественно множеству аксиом. На рисунке приняты следующие сокращения:

- КТ – контрольные точки;
- ПрП – проблемное пространство;

Рис.4. Схема онтологии задач

- ТС – толковый словарь;
- Т-О, Т-П – термины-объекты и термины-процессы;
-
- КТ1 – анализ полноты сформированных списков терминов;
- КТ2 – анализ полноты сформированных списков задач, процедур и методов;
- КТ3 (КТ7, КТ11) – выполняется анализ “Список исчерпан?”;
- КТ4 (КТ8, КТ12) – выполняется анализ “Отношения установлены для всех пар [,]?”;
- КТ5 (КТ9, КТ13) – выполняется анализ “Функции интерпретации определены для всех ?”;
- КТ6 (КТ10, КТ14) – анализ полноты и непротиворечивости онтологии объектов (процессов, задач) экспертом.

Рис.5. Граф-схема алгоритма проектирования онтологий

Заключение. Выполнен анализ особенностей знаниеориентированных информационных систем. На его основе предложен общий подход к построению онтологической базы знаний, функционирующей в составе онтологоуправляемой информационной системы, который назван системно-онтологическим. Компонентами онтологической базы знаний являются онтология объектов, онтология процессов и онтология задач, для которых разработаны схемы моделей и методика проектирования. СОП и соответствующая технология его реализации предполагают анализ ПдО, извлечение, представление и обработку предметных знаний. Вопрос автоматизации приобретения знаний, в частности из множества естественно-языковых объектов, остаётся по-прежнему актуальным.

1. Толковый словарь по искусственному интеллекту / Авт.-сост. А.Н. Аверкин, М.Г. Гаазе-Рапопорт, Д.А. Поспелов. – М.: Радио и связь, 1992. – 256 с.
2. Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. and Lorensen, W. (1991). *Object-oriented modeling and design*. Englewood Cliffs, New Jersey: Prentice Hall.
3. [Natalya F. Noy](#) and [Deborah L. McGuinness](#). "Ontology Development 101: A Guide to Creating Your First Ontology". Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001. – available at http://protege.stanford.edu/publications/ontology_development/ontology101.html.
4. Gruber, T. R. 1993. A translation approach to portable ontology specifications // *Knowledge Acquisition*, 5 (2): pp. 199-220.
5. Философско-методологические основания системных исследований: Системный анализ и системное моделирование // Сб. статей, ВНИИ системных исследований. – М.: Наука, 1983. – 324 с.
6. Згуровский М.З., Панкратова Н.Д. Системный анализ: проблемы, методология, приложения. – К.: Наук. думка, 2005. – 744 с.
7. Добров Б.В., Лукашевич Н.В., Невзорова О.А., Федунев Б.Е. Методы и средства автоматизированного проектирования прикладной онтологии // Изв. РАН. Теория и системы управления. М.: 2004. – № 2. – С.58-68.
8. Невзорова О.А. Онтолингвистические системы: методологические основы построения // Научная сессия МИФИ-2007. Сборник научных трудов. Том 3. Интеллектуальные системы и технологии. М., 2007. – С. 84-85.
9. Методический комплекс по дисциплине "Современные проблемы науки" // Методология анализа предметных знаний. – Доступно в Интернете: http://old.ulstu.ru/people/SOSNIN/umk/Modern_Scientific_Problems/beloborodov/item_znan.htm.
10. Палагин О., Кургаев О. Міждисциплінарні наукові дослідження: оптимізація системно-інформаційної підтримки // Вісник НАН України. – 2009. – № 3. – С. 14-25.
11. Марка Д. А., МакГоуэн К. Методология структурного анализа и проектирования. – М.: "МетаТехнология", 1993. – 239 с.
12. Методология функционального моделирования IDEF0. РД IDEF0 – 2000 // Госстандарт России. – Москва. – 2000. – 75 с. – Доступно на www.nsu.ru/smk/files/idef.pdf.
13. IDEF5 Method Report. – Armstrong Laboratory AL/HRGA Wright-Patterson Air Force Base, Ohio 45433. – 187p. – available at www.idef.com/pdf/Id5.pdf.
14. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. – СПб.: Питер, 2001. – 384 с.
15. Палагин А.В., Яковлев Ю.С. Системная интеграция средств компьютерной техники. – Винница: «УНІВЕРСУМ-Вінниця», 2005. – 680 с.
16. A. Gomez-Perez, M. Fernandez-Lopez, and O. Corcho, *Ontological engineering*. Springer, 2004.
17. Клещев А.С., Артемьева И.Л. Отношения между онтологиями предметных областей. Ч.1. // Информационный анализ, Выпуск 1, С. 2, 2002. – С.4–9.
18. Андреев А.М., Березкин Д.В., Симаков К.В. Особенности проектирования модели и онтологии предметной области для поиска противоречий в правовых электронных библиотеках. – Доступно на www.inteltec.ru/publish/articles/textan/RCDL2004.shtml.
19. Андон Ф.И., Яшунин Л.Е., Резниченко В.И. Логические модели интеллектуальных информационных систем. – К.: Наук. думка, 1999. – 397 с.

20. Войшвилло Е.К. Понятие. – М.: МГУ, 1967. – 286 с.
21. Ивлев Ю.В. Логика: учебник для вузов. – М.: «Логос». – 1997. – 272с.
22. Sowa, J. F. Conceptual Graphs as a universal knowledge representation// In: Semantic Networks in Artificial Intelligence, Spec. Issue of An International Journal Computers & Mathematics with Applications. (Ed. F. Lehmann), Vol.23, Number 2–5, 1992, Part 1, – p.75–95.
23. SUO, (2001), The IEEE Standard Upper Ontology web site, <http://suo.ieee.org>.
24. Niles I., Pease A. Towards a Standard Upper Ontology. //In proceeding of the 2nd International Conference on Formal Ontology and Information Systems (FOIS-2001), Welty C., and Smith B., eds., Ogunquit, Maine. – October, 17-19, 2001. – 8pp. – available at <http://home.earthlink.net/~adampease/professional/FOIS.pdf>.
25. The Mikrokosmos Ontology. – Available at <http://crl.nmsu.edu/Research/Projects/mikro /htmls/asis.paper-htmls/node4.html>.
26. Палагін О.В., Петренко М.Г. Модель категоріального рівня мовно-онтологічної картини світу // Математичні машини і системи. – 2006. – №3. – С. 91–104.