

Виталий Литвинов, Ирина Посадская

ПРОЕКТИРОВАНИЕ ФОРМАЛИЗОВАННОГО ПРЕДСТАВЛЕНИЯ ПРЕДМЕТНОЙ ОБЛАСТИ ИСХОДЯ ИЗ ЕЕ ТЕКСТОВО-ГРАФИЧЕСКОГО ОПИСАНИЯ

Vitalii Lytvynov, Iryna Posadska

ПРОЕКТУВАННЯ ФОРМАЛІЗОВАНОГО ПРЕДСТАВЛЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ВИХОДЯЧИ З ЇЇ ТЕКСТОВО-ГРАФІЧНОГО ОПИСУ

Vitalii Lytvynov, Iryna Posadska

DOMAIN FORMALIZED PRESENTATION DESIGN BASED ON ITS TEXT AND GRAPHIC DESCRIPTIONS

Обоснована важность формализации предметной области курса не в виде параграфов и разделов документа, а в виде тех средств, которые дают возможности представлять знания в системе и использовать их дальше в процессе обучения. Идентификацию объектов предметной области предлагается выполнять с помощью подхода подчеркивания существительных. Предложены элементарные операции и стратегии, с помощью которых создается костяк формализованного представления предметной области. Предложен жизненный цикл разработки формализованного представления учебного курса.

Ключевые слова: автоматизированная система обучения, инженер по знаниям, эксперт в предметной области, формализация, элементарные операции, стратегия.

Рис.: 1. Табл.: 2. Библ.: 13.

Обґрунтовано важливість формалізації предметної області курсу не у вигляді параграфів і розділів документа, а у вигляді тих засобів, які дають можливості представляти знання в системі і використовувати їх далі в процесі навчання. Ідентифікацію об'єктів предметної області пропонується виконувати за допомогою підходу підкреслення іменників. Запропоновано елементарні операції і стратегії, за допомогою яких створюється кістяк формалізованого представлення предметної області. Запропоновано життєвий цикл розробки формалізованого представлення навчального курсу.

Ключові слова: автоматизована система навчання, інженер зі знань, експерт у предметній області, формалізація, елементарні операції, стратегія.

Рис.: 1. Табл.: 2. Бібл.: 13.

The article substantiates course domain formalization importance not in the form of paragraphs and chapters of the document, but as the means that make it possible to represent knowledge in the system and use them during the learning process. Identification of domain's objects is proposed to perform by using the nouns underline approach. Elementary operations and strategies, by means of which the foundation of domain formalized representation is created, are proposed. The life-cycle development of learning course formalized representation is proposed.

Key words: automated learning system, knowledge engineer, domain expert, formalization, elementary operations, strategy.

Fig.: 1. Tabl.: 2. Bibl.: 13.

Постановка проблемы. Практически все современные автоматизированные системы обучения (АСО) используют частично структурированную информацию о предметной области, представленную в виде текстово-графических материалов, представленных в виде параграфов, разделов и подразделов документа. Такого рода формализация не позволяет дальше использовать знания в процессе обучения.

На сегодня актуальными задачами в создании автоматизированных систем являются: дополнение учебных курсов формализованными представлениями фрагментов предметной области; выбор инструментального средства для формализации; определение стратегии процесса формализации знаний; интеграция формализованных знаний о предметной области с текстово-графическим представлением раздела документа.

Анализ последних исследований и публикаций. В большинстве форм представлений знаний центральной идеей является фиксация множества концептов и отношений между ними [1], которые хорошо сочетаются со структурными моделями.

Структурные модели, как правило, носят статический характер, фактор времени в них не играет существенной роли. Для описания таких моделей разработано множество специализированных формализмов задания динамики поведения систем, к которым относятся системы дифференциальных уравнений, конечные автоматы [2], марковские и

полумарковские процессы [3], агрегативные модели [4], сети Петри [5], логико-динамические модели [6] и др. Все эти средства описания динамики в явной или неявной форме имеют дело с множеством состояний системы и переходами между состояниями. Они используются как для описания фактов, так и знаний, задающих классы. Модели, отражающие динамику поведения, часто называют динамическими моделями.

Динамические модели характеризуются наличием множества состояний и возможных переходов между ними во времени. Исходя из этого, в перечень задач, решаемых с помощью динамических моделей, входят задачи прогноза или достижимости состояний, задачи допустимости траекторий поведения, задачи определения моментов обрыва траектории и т. д.

Однако универсального способа формализации предметной области курса в АСО не существует. Например, в работе [7] проведен сравнительный анализ моделей и сделан выбор в пользу фреймов, за счет того, что структура фрейма не привязывается к особенностям предметной области, а определяется свойствами моделируемых понятий.

При таком большом наличии способов формализации, можно сделать вывод, что выбор способа не так важен, как выбор наиболее подходящего инструментального средства для представления знаний. В нашем подходе предлагается использовать наиболее известный формализованный язык Unified Modelling Language (UML).

Целью статьи является обоснование нового подхода к формализации предметной области курса в автоматизированных системах обучения, который базируется на применении формализованного языка UML.

Изложение основного материала. Как известно [8], UML представляет собой многоязыковую систему с возможностями задания структурных моделей (диаграммы классов), сценариев взаимодействия элементов моделей (диаграммы последовательностей и диаграммы взаимодействия объектов), задания динамики функционирования системы (диаграммы последовательностей, диаграммы состояний), взаимодействия параллельных процессов (диаграммы активностей) и др. Указанные выше диаграммы играют решающую роль в формализованном представлении предметной области курса; и, как показали исследования в области программной инженерии [9], могут быть получены из текстового описания объекта формализации на естественном языке.

Архитектура автоматизированной системы обучения (АСО) состоит из трех функциональных модулей: модуль формирования знаний о предметной области, модули обучения и контроля [10].

Главной функцией модуля формирования знаний о предметной области является *функция структуризации знаний*. Известно, что знания в АСО могут быть представлены в естественно-языковой текстово-графической форме и формализованной. Не все знания подлежат формализации. В данном функциональном модуле задействованы две категории пользователей системы – эксперт в предметной области и инженер по знаниям.

Главным направлением работы эксперта в предметной области является подготовка курса по предметной области, а именно выполнение следующих базовых функций:

- отбор и выбор материалов, которые необходимо изучить;
- разработка сопровождающих иллюстративных материалов;
- разработка таблиц, графиков и других вспомогательных учебных материалов;
- представление знаний по предметной области в текстовой форме – первичная структуризация, т. е. разбиение знаний на разделы, подразделы, параграфы и т. д.;
- выделение фрагментов предметной области, которые подлежат формализации;
- определение степени покрытия предметной области формализованными представлениями.

TECHNICAL SCIENCES AND TECHNOLOGIES

Инженер по знаниям приступает к работе, когда первичная структуризация уже проведена. Анализируя результаты работы эксперта в предметной области, выделяются те фрагменты предметной области, которые подлежат формализации.

Среди основных функций инженера по знаниям выделяют следующие:

- выбор стратегии формализации представленных компонент знаний;
- выделение фрагментов предметной области, которые подлежат формализации;
- формализация выделенных представленных компонент знаний;
- установление прямых ссылок между формализованными знаниями и структурированными текстово-графическими и наоборот;
- поддержка знаний в актуальном состоянии.

Его основной функцией является формализация знаний о предметной области. Выбор способа формализации является важнейшей задачей.

Как видим, работа над формализованным представлением предметной области исходя из ее текстово-графического описания зависит от эксперта в предметной области и инженера по знаниям (рис.).

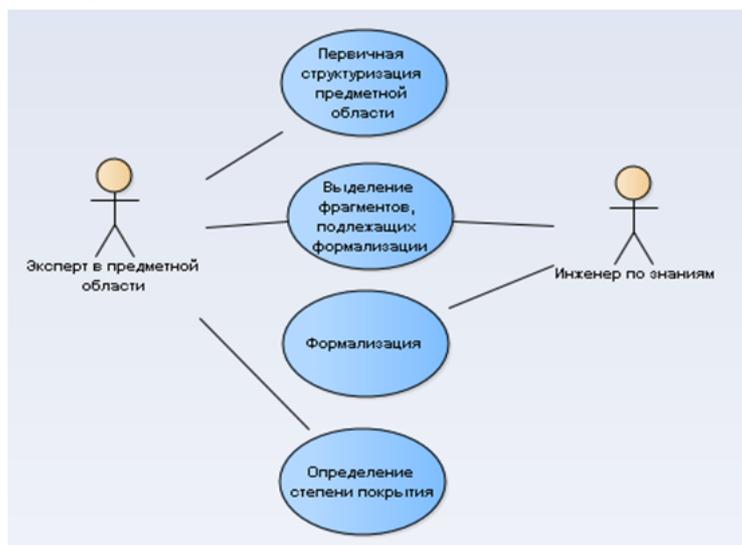


Рис. Диаграмма прецедентов (Use Case) взаимодействия эксперта в предметной области и инженера по знаниям при формализации предметной области

Для построения формализованной модели предметной области эксперт и инженер используют ряд разработанных подходов к выделению объектов и связей [11]:

- Подход подчеркивания существительных.
- Идентификация причинных объектов.
- Идентификация сервисов (пассивные объекты).
- Идентификация на базе объектов реального мира.
- Идентификация объектов на базе физических устройств.
- Идентификация ключевых концептов.
- Идентификация объектов на базе транзакций.
- Идентификация сохраняемой информации.
- Идентификация визуальных элементов.
- Использование сценариев.
- Идентификация связей объектов.

Перечисленные выше подходы выделения объектов и связей могут использоваться как отдельно, так и в совокупности при текстовом анализе. Однако следует отметить слабую степень формализации данного процесса как в отношении построения классовых диаграмм, так и в отношении диаграмм, описывающих сценарии поведения.

При работе с естественно-языковым представлением предметной области чаще всего используется подчеркивание существительных. Определение потенциальных объектов осуществляется путем подчеркивания каждого существительного или фраз с существительными в тексте. Таким образом, формируется словарь потенциальных объектов предметной области.

Претенденты, идентифицированные подобным образом, могут попасть в одну из четырех категорий:

- объекты, представляющие интерес;
- актеры;
- объекты, не представляющие интереса;
- атрибуты объектов.

Особенности данного подхода:

1. Актеры, обычно уже идентифицированы в модели прецедентов, но иногда обнаруживаются некоторые новые актеры, которые не имеют прямого отношения к нашей системе.

2. В описании проблемы атрибуты также представлены как существительные.

3. Иногда атрибут может трактоваться как свойство объекта.

4. Если имеется сомнение, необходимо классифицировать существительное как объект.

5. Если последующий анализ показывает, что объект не является существенно интересным, тогда он может быть включен как атрибут в некоторый другой объект.

Не смотря на то, что этот метод является наиболее распространенным, и для него степень формализации предметной области не является оптимальной.

Развивая дальнейшую формализацию данного метода, необходимо решить следующие задачи:

1. Задачу обработки естественно-языковых текстов с целью выделения объектов, атрибутов, классов объектов и отношений между ними;

2. Задачу интеграции классовых диаграмм;

3. Задачу оценки показателей качества классовых диаграмм;

4. Задачу эквивалентных преобразований классовых диаграмм;

5. Задачу согласования классовых диаграмм с диаграммами последовательностей и диаграммами состояния, описывающих поведение объектов.

Для примера выполнения первой задачи рассмотрим **обобщенный алгоритм обработки текста для формализации предметной области в терминах UML**:

1. Выделение из текста T множеств словосочетаний, обозначающих группы существительного N_t образующих словарь первичных имен сущностей.

2. Переработка N_t и текста T с целью избавления от синонимов и антонимов. Последние переименовываются, а для синонимов в словаре оставляется единственный. В результате образуется словарь N'_t и текст T' .

3. Группирование предложений текста T' на терминологические группы G_n , где $n \in N'_t$. В принципе, предложение текста T' может включаться в несколько групп.

4. Сопоставление каждому элементу $n \in N'_t$ информации о количестве сущностей в системе, соответствующей текстовому описанию T' .

Таким образом, определена функция $F: N'_t \rightarrow L$. Здесь в качестве элементов множества L могут выступать целые положительные числовые значения и неопределенность *.

5. На элементах m множества N'_t выделим атрибутные связи r_m , которые определяются следующим образом: сущность $n \in N'_t$ является атрибутом сущности $m \in N'_t$, $((n, m) \in r_m)$, если n не содержит других сущностей и ее можно рассматривать, как свойство сущности m . Выделение атрибутных связей рационально вести с использованием онтологий или словарей - глоссариев по данной предметной области.

Множество атрибутов множества N'_t обозначим N_t^a , а множество атрибутивных связей – R_a .

6. На элементах m множества N'_t выделим связи по обобщению, которые рассматриваются, как антипод специализации. Они определяются следующим образом: сущность $n \in N'_t$ является прямым обобщением сущности $m \in N'_t$, если n является обобщением m и не существует отличной от n и m сущности l , которая является обобщением m и для которой n является обобщением. Множество прямых связей по обобщению обозначим через R_o . Выделение связей по обобщению рационально вести с использованием онтологий или словарей – глоссариев по данной предметной области.

7. На элементах m множества N'_t выделим агрегативные связи $R_{ag} = \{r_n^m\}$, которые рассматриваются, как вхождение одной сущности $n \in N'_t$ в другую $m \in N'_t$, как элемента. При этом различаются два вида агрегативных связей: связи, для которых объекты сущности n порождаются, уничтожаются и не могут существовать вне породившего объекта m (композиционные связи) и объекты с временным входением в состав других объектов. Множества таких связей обозначаются R_{ag}^1 и R_{ag}^2 соответственно, $R_{ag} = R_{ag}^1 \cup R_{ag}^2$.

Сущности n и m не должны иметь атрибутивной связи. В терминологически группах G_n и G_m должны иметься предположно-глагольные формы, отражающие отношение включения n в m .

8. Для предложений каждой терминологической группы G_n , $n \in N'_t$ могут иметься глагольно-предложные формы, отражающие отношения следования событий, в которых участвуют объекты сущностей n и m . Эти связи носят название ассоциативных связей r_{nm} , а множество ассоциативных связей обозначается как $R_{as} = \{r_{nm}\}$. Будем различать два вида ассоциативных связей: однонаправленные, трактуемые как то, что событие, в котором участвует n , предшествует событию, в котором участвует m , и двунаправленные, трактуемые как две однонаправленные связи, действующие в противоположных направлениях.

9. Другие виды связей определяемые глагольно-предложными формами для сущностей n и m в предложениях терминологической группы G_n , $n \in N'_t$ или для предметной области не имеют значения, или относятся к категории «зависимость».

Описанный выше подход к построению моделей предметных областей обладает существенными недостатками:

- он тяжело поддается автоматизации;
- он рассчитан на работу с небольшими по объему текстами;
- помимо собственно текста он требует привлечения дополнительного материала, такого как глоссарии, терминологические словари, онтологии.

При этом привлечение эксперта в предметной области является непременным условием успеха. Максимум, на что можно рассчитывать при этом подходе - это на создание костяка формализованного представления предметной области или ее небольших фрагментов. Вместе с тем для создания реальных формализованных предметных областей требуется как привлечение множества экспертов, так и выделение необходимого времени.

Это может быть сделано за счет:

- введения элементарных операций преобразования формализованных схем;
- использования стратегий, их постепенного формирования за счет детализации или обобщения;
- применения операций интеграции формализованных фрагментов предметных областей в единое целое.

При такой постановке первым шагом является введение множеств элементарных операций диаграмм. Любую элементарную операцию можно рассматривать как функцию f , определенную на множестве диаграмм (схем) S в множество схем S , $f: S \rightarrow S$.

В каком-то смысле наборы элементарных операций по формированию UML диаграмм в зачаточном виде имеются в любых объектно-ориентированных CASE системах [12].

При построении ER моделей в [13] также используются элементарные операции. В данном подходе при построении формализованного представления предметной области с ориентацией на UML мы введем два вида элементарных операций:

- элементарная операция «сверху-вниз»;
- элементарная операция «снизу-вверх».

Первые рационально использовать при построении предметной области проводя детализацию сверху-вниз. Вторые – наоборот, при построении предметной области снизу-вверх, за счет введения обобщений и составных сущностей.

Элементарные операции «сверху-вниз» характеризуются следующими свойствами:

- Начальная схема представляет собой единственный концепт, а результирующая состоит из небольшого набора концептов.
- Все имена концептов преобразуются в новые имена, описывающие исходный концепт на более низком абстрактном уровне.
- Логические связи должны наследоваться единственным концептом результирующей схемы.

Приведем возможный набор элементарных операций «сверху-вниз» для преобразования классовых UML диаграмм, описывающих предметную область.

1. Группа элементарных операций T_1 , преобразует сущность в отношение между двумя сущностями.

Операции этой группы могут образовывать отношения следующих типов:

- двунаправленная ассоциация, с кардинальными числами или без них $T_{1,1}, T_{1,2}$;
- однонаправленная ассоциация с кардинальными числами или без них $T_{1,3}$;
- прямая или обратная агрегация $T_{1,4}, T_{1,5}$;
- прямое или обратное обобщение $T_{1,6}, T_{1,7}$;
- прямое или обратное композитное отношение $T_{1,8}, T_{1,9}$.

2. Группа элементарных операций $T_{2,1}(n)$, детализирующая сущность в иерархию из $n+1$ сущностей по отношению обобщения.

3. Группа элементарных операций $T_{3,1}(n)$ разбивает сущность на множество из n независимых сущностей.

4. Группа элементарных операций $T_{4,1}(n)$ разбивает отношение ассоциации между двумя сущностями на два и более отношений (n) между теми же сущностями.

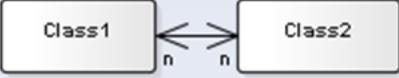
5. Группа элементарных операций $T_{5,1}(n)$ преобразует отношение ассоциации в ненаправленный ассоциативный путь между теми же сущностями, проходящий между новыми n промежуточными сущностями.

6. Элементарная операция $T_{6,1}$ вводит в сущность ее атрибуты.

Графическое представление элементарных операций «сверху-вниз» дано в табл. 1.

Таблица 1

Элементарные операции «сверху-вниз»

№ п/п	Группа	Элементарные операции	Входной фрагмент	Результат
1	2	3	4	5
1	T_1	$T_{1,1}$		
2	T_1	$T_{1,2}$		

Окончание табл. 1

1	2	3	4	5
3	T ₁	T _{1,3}		
4	T ₁	T _{1,4}		
5	T ₁	T _{1,5}		
6	T ₁	T _{1,6}		
7	T ₁	T _{1,7}		
8	T ₁	T _{1,8}		
9	T ₁	T _{1,9}		
10	T ₂	T _{2,1(n)}		
11	T ₃	T _{3,1(n)}		
12	T ₄	T _{4,1(n)}		
13	T ₅	T _{5,1(n)}		
14	T ₆	T _{6,1}		

Указанные выше элементарные операции «сверху-вниз» не позволяют создать все допустимые классовые диаграммы.

Элементарные операции «снизу-вверх» вводят новые концепты и свойства, которые отсутствовали в предыдущих версиях диаграммы, или модифицируют некоторые имевшиеся в диаграмме концепты.

Элементарные операции «снизу-вверх» используются при проектировании диаграммы, когда мы раскрываем те особенности предметной области, которые не обна-

ружены на любом уровне абстракции предыдущей версии диаграммы. Они используются также тогда, когда диаграмма преобразуется в более общую схему.

Приведем возможный набор элементарных операций «снизу-вверх» для преобразования классовых UML диаграмм, описывающих предметную область.

1. Элементарная операция B_1 порождает в диаграмме новую сущность. Она употребляется, когда инженер по знаниям обнаружил новый концепт со специфическими свойствами, который не появлялся в предыдущей диаграмме.

2. Группа элементарных операций B_2 порождает новое отношение между ранее определенными сущностями диаграммы. В зависимости от вида порождаемого отношения различаются:

- элементарная операция $B_{2,1}$ для отношения двунаправленной ассоциации;
- элементарная операция $B_{2,2}$ для отношения прямой однонаправленной ассоциации;
- элементарная операция $B_{2,3}$ для отношения обратной однонаправленной ассоциации;
- элементарная операция $B_{2,4}$ для прямого агрегативного отношения;
- элементарная операция $B_{2,5}$ для обратного агрегативного отношения;
- элементарная операция $B_{2,6}$ для прямого композитного отношения;
- элементарная операция $B_{2,7}$ для обратного композитного отношения.

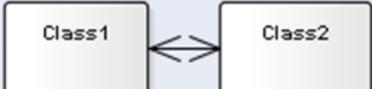
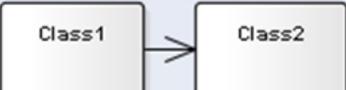
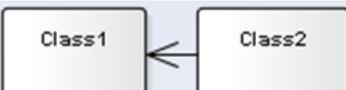
3. Элементарная операция B_3 порождает новую сущность, которая рассматривается как обобщение (вершина иерархии) для ранее определенных в диаграмме сущностей.

4. Элементарная операция B_4 порождает новый атрибут для ранее определенной сущности.

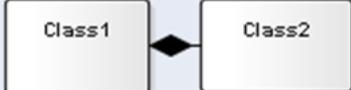
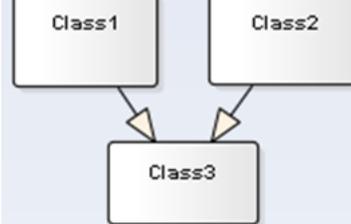
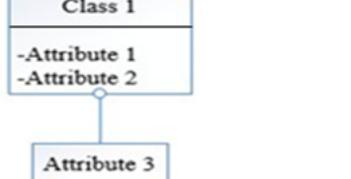
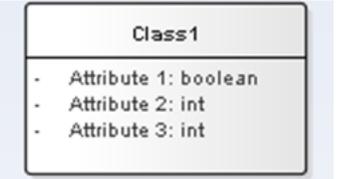
Графическое представление элементарных операций «снизу-вверх» дано в табл. 2.

Таблица 2

Элементарные операции «снизу-вверх»

№ п/п	Группа	Элементарная операция	Входной фрагмент		Результат
1	2	3	4		5
1	B_1	B_1			
2	B_2	$B_{2,1}$			
3	B_2	$B_{2,2}$			
4	B_2	$B_{2,3}$			
5	B_2	$B_{2,4}$			
6	B_2	$B_{2,5}$			
7	B_2	$B_{2,6}$			

Окончание табл. 2

1	2	3	4	5
8	B ₂	B _{2,7}		
9	B ₃	B _{3,1}		
10	B ₄	B _{4,1}		

Стратегии, используемые при построении формализованных моделей предметных областей на базе UML. Общие принципы использования элементарных операций при построении моделей предметной области назовем стратегиями построения предметной области.

Для построения классовых диаграмм, описывающих предметную область, будем различать четыре стратегии:

- «Сверху-вниз»;
- «Снизу-вверх»;
- «Изнутри-наружу»;
- Смешанная.

Каждая из них характеризуется использованием определенных элементарных операций.

Стратегия «сверху-вниз». При использовании стратегии «сверху-вниз», классовая диаграмма, описывающая предметную область, получается путем детализации за счет использования только элементарных операций «сверху-вниз». Каждое использование операции добавляет новые детали в диаграмму. На каждом шаге преобразования происходит детализация одной сущности или связи, в то время как остальная часть диаграммы остается неизменной.

Таким образом, процесс построения предметной области, представленной в виде классовой UML диаграммы, в случае использования стратегии «сверху-вниз» представляется в виде последовательности применений к схеме диаграммы элементарных операций «сверху-вниз»: $\langle T^1, T^2, \dots, T^n \rangle$, где $T^i \in T, i = 1, \dots, n$, причем исходная диаграмма, S_0 , обрабатываемая T^1 , состоит из единственной сущности, а конечная классовая диаграмма S_n , получаемая в результате применения операции. T^n – есть искомое формализованное представление предметной области. При этом $S_0 < S_1 < S_2 \dots < S_n$.

Таким образом, стратегия «сверху-вниз» наиболее применима, когда предметная область строится, начиная с наиболее абстрактных понятий. Она позволяет нам «видеть лес, а не отдельные деревья».

Стратегия «снизу-вверх». При использовании стратегии «снизу-вверх», классовая диаграмма, описывающая предметную область, получается путем детализации за счет использования только элементарных операций «снизу-вверх». Процесс начинается с

элементарных концептов, на базе которых с использованием операций строятся более сложные концепты, и т. д.

Если процесс построения формализованного представления предметной области рассматривать, как последовательность преобразований, выполняемых над диаграммами с помощью элементарных операций «снизу-вверх» $\langle B^1, B^2, \dots, B^n \rangle$, где $B^i \in B, i = 1, \dots, n$, то здесь на вход B^1 поступает диаграмма, состоящая из низкоуровневых сущностей, имеющих отображение в первичном словаре терминов-существительных, а конечная классовая диаграмма S_n , получается в результате применения элементарной операции B^n . T^n – есть искомое формализованное представление предметной области. При этом $S_0 < S_1 < S_2 \dots < S_n$.

Постепенное добавление абстрактных сущностей может привести к частичной реструктуризации диаграммы за счет переноса свойств сущностей по пути, определяемому отношениями обобщения.

Последовательное применение элементарных операций «снизу-вверх» может привести к образованию на диаграмме множества кластеров связанных сущностей, между которыми в дальнейшем можно установить отношения.

Преимуществом стратегии «снизу-вверх» является простота построения кластеров, что дает возможность быстро формировать прототипы предметных областей.

Недостатками стратегии «снизу-вверх» является трудность интеграции фрагментов сложных диаграмм и необходимость частой реструктуризации диаграммы. Стабилизация состава сущностей и отношений в диаграмме наступает только в конце процесса проектирования.

В случае использования стратегии «снизу-вверх» мы скорее «видим отдельные деревья, а не лес».

Стратегия «изнутри-наружу». Используя стратегию «изнутри-наружу» при проектировании классовой диаграммы для формализованного представления предметной области, прежде всего, выбираются наиболее важные и наиболее очевидные концепты (сущности). Зафиксировав такие концепты, необходимо найти другие концепты, которые являются наиболее концептуально близкими к первым. Далее процесс повторяется. В ER моделировании предметных областей этот процесс носит название «процесса распространения масляного пятна» [13].

Таким образом, стратегия «изнутри-наружу» может рассматриваться как частный случай стратегии «снизу-вверх».

При использовании стратегии «изнутри-наружу» процесс детализации аналогичен подходу «снизу-вверх». Уровни абстракции концептов, представленных в соседних версиях диаграммы аналогичны. Однако потеряны преимущества процессирования, начиная с абстрактных уровней.

Смешанная стратегия. Главной идеей смешанной стратегии является то, что когда предметная область является достаточно сложной, инженер по знаниям делит ее текстовое представление на части, которые далее рассматриваются отдельно.

Одновременно разработчик строит скелет диаграммы, который содержит наиболее важные концепты и включает связи между ними. Наличие скелета диаграммы облегчает интеграцию различных поддиаграмм.

Описанные выше стратегии проектирования формализованного представления предметной области позволяют сделать определенные выводы о характере области применимости данных стратегий.

1. Стратегии «сверху-вниз» и «снизу-вверх» применимы для проектирования небольших предметных областей.

2. Стратегия «изнутри-наружу» применима для проектирования формализованных представлений предметных областей средней размерности.

3. Смешанные стратегии вполне применимы для проектирования формализованных представлений предметных областей учебных курсов.

Однако в последнем случае требуется учесть и тот факт, что отдельные фрагменты общей большой классовой диаграммы могут проектировать разные люди.

С учетом высказанных выше замечаний можно представить жизненный цикл разработки формализованного представления учебного курса следующим образом:

1. Анализ текста курса T_1 и разбиение его на разделы $T_{1,1}, T_{1,2}, \dots, T_{1,n}$.

2. Составление скелета формализованного представления модели предметной области S_{sk} .

3. Разработка формализованных представлений фрагментов предметной области S_1, S_2, \dots, S_m .

4. Интеграция фрагментов формального описания предметной области в общее формальное описание S .

5. Анализ качества формального описания модели предметной области S и ее корректировки с целью улучшения качества.

Эта разновидность жизненного цикла опирается на смешанную стратегию разработки, о чем свидетельствуют этапы 2, 3, 4.

Выводы. Предложенный подход к формализации предметной области курса в АСО базируется на применении формализованного языка UML. На базе подхода подчеркивания существительных выделяются объекты (классы и отношения), описывающие формализованное представление предметной области, составляются терминологические словари, глоссарии и т.п..

Результатами выполнения перечисленных выше трех первых этапов жизненного цикла разработки формализованного представления учебного курса являются: терминологический словарь для сущностей предметной области курса (включая предметный указатель) L_1 ; терминологический словарь для глагольных и предложных форм в текстовом представлении курса L_2 ; стандартизованное представление текста курса, свободное от синонимов, антонимов и переработанное на предмет стандартизации синтаксических структур T_2 ; имеющиеся онтологические описания предметной области курса S^0 ; скелет S_{sk} диаграммы предметной области курса, содержащий множество наиболее важных концептов и связей между ними: $S_{sk} = \langle N_{sk}, R_{sk} \rangle$; классовая поддиаграмма S_i для каждого $i \in N_{sk}$.

В процессе создания фрагментов и общей схемы предметной области курса формируются прямые ссылки на оглавление, разделы, подразделы и параграфы текстового представления курса. А также обратные ссылки от словарей терминов, глоссариев на формальное представление предметной области или ее фрагмента. Это дает возможность использовать формальную структуризацию курса для формирования контрольных вопросов и более точно указывать фрагменты текстового представления знаний для адаптации материала при изучении и повторении под уровень знаний обучаемого.

Список использованных источников

1. *Поспелов Д. А.* Представление знаний. Опыт системного анализа / Д. А. Поспелов // Системные исследования. Методологические проблемы. – 1986. – № 17. – С. 83–102.

2. *Глушков В. М.* Абстрактная теория автоматов / В. М. Глушков // Успехи математических наук. – 1961. – № 6 (101). – С. 3–62.

3. *Королюк В. С.* Процессы марковского восстановления в задачах надежности систем / В. С. Королюк, А. Ф. Турбин. – К. : Наук, думка, 1982. – 236 с.

4. *Бусленко Н. П.* Лекции по теории сложных систем / Н. П. Бусленко, В. В. Калашников, И. Н. Коваленко. – М. : Сов. радио, 1973. – 440 с.

5. Питерсон Дж. Теория сетей Петри и моделирование систем / Дж. Питерсон. – М. : Мир, 1984. – 264 с.
6. Жук К. Д. Исследование структур и моделирование логико-динамических систем : монография / К. Д. Жук, А. А. Тимченко, Т. И. Доленко. – К. : Наукова думка, 1975. – 199 с.
7. Любченко В. В. Модели знаний для предметных областей учебных курсов / В. В. Любченко // Искусственный интеллект. – 2008. – № 4. – С. 458–462.
8. Bruce Powel Douglass. Real – Time UML. Second Edition. Developing Efficient Objects for Embedded Systems / Bruce Powel Douglass. – Wesley, 1999. – 238 p.
9. Коммервилл И. Инженерия программного обеспечения / И. Коммервилл. – М. : Вильямс, 2002. – 624 с.
10. Литвинов В. В. Архитектура знание-ориентированной автоматизированной системы обучения / В. В. Литвинов, И. С. Посадская, М. В. Савельев // Технічні науки та технології. – 2016. – № 3 (5). – С. 122–130.
11. Об'єктно-орієнтоване моделювання при проектуванні вбудованих систем і систем реального часу / В. В. Литвинов, С. В. Голуб, К. М. Григор'єв, В. Ю. Жигульська. – Черкаси : Черкаський національний університет ім. Б. Хмельницького, 2011. – 379 с.
12. Трофимов С. А. CASE-технологии: практическая работа в Rational Rose / С. А. Трофимов. – Изд. 2-е. – М. : Бинوم-Пресс, 2002. – 288 с.
13. Carlo Batini Conceptual database design: an entity-relationship approach / Carlo Batini, Stefano Ceri, Shamkant B. Navathe. – The Benjamin/Cummings Publishing Company, Inc. 1992. – 490 p.

References

1. Pospelov, D.A. (1986). Predstavlenie znaniy. Opyt sistemnogo analiza [Knowledge representation. Experience in System Analysis]. *Sistemnye issledovaniya. Metodologicheskie problem – System Research. Methodological problems*, no. 17, pp. 83–102 (in Russian).
2. Glushkov, V.M. (1961). Abstraktnaya teoriya avtomatov [The abstract theory of automatic machines]. *Uspekhi matematicheskikh nauk – Successes of Mathematical Sciences*, no. 6 (101), pp. 3–62 (in Russian).
3. Korolyuk, V.S. & Turbin, A.F. (1982). *Protsessy markovskogo vosstanovleniya v zadachakh nadezhnosti system [Markov renewal processes in systems reliability problems]*. Kiev: Nauk. dumka (in Russian).
4. Buslenko, N. P., Kalashnikov, V.V., Kovalenko, I.N. (1973). *Lektsii po teorii slozhnykh system [Lectures on the theory of complex systems]*. Moscow: Soviet radio (in Russian).
5. Piterson, Dzh. (1984). *Teoriua seteu Petri i modelirovanie sistem [The theory of Petri nets and modeling systems]*. Moscow: Mir (in Russian).
6. Zhuk, K.D., Timchenko, A.A., Dolenko, T.I. (1975). *Issledovanie struktur i modelirovanie logiko-dinamicheskikh system [Research of structures and modeling of logic-dynamic systems]*. Kiev: Naukova dumka (in Russian).
7. Lyubchenko, V.V. (2008). Modeli znaniy dlya predmetnykh oblastey uchebnykh kursov [Knowledge Models for the learning courses domains]. *Iskusstvennyi intellekt – Artificial Intelligence*, no. 4, pp. 458–462 (in Russian).
8. Douglass, Bruce Powel (1999). *Real - Time UML. Second Edition. Developing Efficient Objects for Embedded Systems*. Wesley.
9. Sommervill, I. (2002). *Inzheneriia programmnoho obespecheniia [Software Engineering]*. Moscow: Viliams (in Russian).
10. Litvinov, V.V., Posadskaya, I.S., Savel'yev, M.V. (2016). Arkhitektura znaniye-orientirovannoy avtomatizirovannoy sistemy obucheniya [Architecture of knowledge-oriented automated system of learning]. *Tekhnichni nauki ta tekhnologii – Technical Sciences and Technologies*, no. 3 (5), pp. 122–130 (in Russian).
11. Lytvynov, V.V., Holub, S.M., Hryhoriev, K.M., Zhyhulska, V.Y. (2011). *Obiektno-oriientovane modeliuvannia pry proektuvanni vbudovanykh system i system realnoho chasu [Object-oriented modeling in the design of embedded systems and real-time systems]*. Cherkasy: Cherkaskyi natsionalnyi universytet im. B. Khmelnytskoho (in Ukrainian).

TECHNICAL SCIENCES AND TECHNOLOGIES

12. Trofimov, S.A. (2002). *CASE-tekhnologii: prakticheskaya rabota v Rational Rose [CASE-technology: practical work in Rational Rose]* (2nd ed.). Moscow: Binom-Press (in Russian).
13. Carlo Batini, Stefano Ceri, Shamkant B. Navathe (1992). *Conceptual database design: an entity-relationship approach*. The Benjamin/Cummings Publishing Company, Inc.

Литвинов Виталий Васильевич – доктор технических наук, профессор, заведующий кафедрой информационных технологий и программной инженерии, Черниговский национальный технологический университет (ул. Шевченко, 95, г. Чернигов, 14027, Украина).

Литвинов Віталій Васильович – доктор технічних наук, професор, завідувач кафедри інформаційних технологій та програмної інженерії, Чернігівський національний технологічний університет (вул. Шевченка, 95, м. Чернігів, 14027, Україна).

Lytvynov Vitalii – Doctor of Technical Sciences, Professor, Head of the Department of Information Technologies and Software Engineering, Chernihiv National University of Technology (95 Shevchenka Str., 14027, Chtrnihiv, Ukraine).

E-mail: vlitvin@ukrsoft.ua

ORCID: <http://orcid.org/0000-0001-9622-3871>

Посадская Ирина Сергеевна – ассистент кафедры информационных технологий и программной инженерии, Черниговский национальный технологический университет (ул. Шевченко, 95, г. Чернигов, 14027, Украина).

Посадська Ірина Сергіївна – асистент кафедри інформаційних технологій і програмної інженерії, Чернігівський національний технологічний університет (вул. Шевченка, 95, м. Чернігів, 14000, Україна).

Posadska Iryna – assistant of the Department of Information Technologies and Software Engineering, Chernihiv National University of Technology (95 Shevchenka Str., 14027, Chtrnihiv, Ukraine).

E-mail: irrkin@gmail.com

ORCID: <http://orcid.org/0000-0002-4905-2552>