

А. А. Малых, А. В. Манцивода

## ОНТОЛОГИИ, МЕТАДАННЫЕ И СЕМАНТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

???????????

### Введение

Данная работа может рассматриваться как призыв к более активному использованию семантического программирования [1–2] в распределенных информационных системах. В ней делается попытка дать некоторое обобщение результатов экспериментальных исследований, которые проводились нами в последние 2–3 года. Эти исследования были связаны с приложением логических средств к различным аспектам управления информацией в виртуальных средах. Работа велась по следующим направлениям:

- исследование канонических форматов представления структурированной текстовой информации (включая образовательные и научные документы, обработку естественнонаучных текстов и математических формул), манипулирование структурированными документами как математическими объектами [14–15, 18–22];
- развитие технологии метаописаний информационных ресурсов, в первую очередь ресурсов образовательного характера [7–13, 15–17];
- разработка системы описания знаний о предметных областях, предназначенной для использования автоматическими Интернет-сервисами (например, поисковыми машинами) [9, 23];
- развитие функционально-логических языков, методов веб-программирования и автоматического генерирования информационных ресурсов [5–6, 14].

Обобщение подходов, связанных с обработкой метаданных, описаний предметных областей и классификаций мы определяем на некотором диалекте семантического программирования. Семантическое программирование понимается нами как универсальный подход — формализм, настраиваемый на конкретные предметные области.

Пусть имеется область  $D$  реального мира, которую мы хотим смоделировать и с которой хотим работать. Настройка семантической системы на предметную область осуществляется в несколько шагов.

**Шаг 1.** Фиксируется модель  $\mathfrak{X} = \langle M_1, \dots, M_s; \Omega \rangle$  типов данных, задающая базовые элементы для построения описаний предметной области  $D$ .

**Шаг 2.** Над моделью  $\mathfrak{X}$  строится подходящая наследственно конечная надстройка  $T(\mathfrak{X})$ , элементы которой служат описаниями объектов области  $D$ . Как и в реальной жизни, описания могут быть как более точными, так и менее точными. Это

приводит к пониманию элементов  $T(\mathfrak{R})$  как аппроксимаций объектов реального мира. Кроме того, разные элементы  $T(\mathfrak{R})$  могут задавать разный уровень полноты описаний объектов. Это приводит к отношению аппроксимации  $\sqsubseteq$ , определенному уже на элементах самого множества  $T(\mathfrak{R})$ .

**Шаг 3.** Отражая иерархическую структуру классификации объектов (таксономии), присущей предметной области  $D$ , на базе  $T(\mathfrak{R})$  строится иерархическая система классов как подмножеств  $T(\mathfrak{R})$  специального вида.

**Шаг 4.** Для работы с описаниями объектов в виде элементов надстройки определяются подходящие  $\Delta_0$  и  $\Sigma$ -языки.

Предварительная работа нашей исследовательской группы показала, что такой подход имеет значительное количество нетривиальных приложений в самых разнообразных сферах обработки информации. В данной работе мы изложим лишь основные принципы настройки семантического программирования на задачи обработки информации в глобальных информационных системах и введем набор базовых понятий, необходимых, с нашей точки зрения, для успешного продвижения в этом направлении. Но прежде дадим краткую характеристику текущего положения в мировом информационном пространстве, которое, как нам представляется, определяет довольно заманчивые перспективы для использования логических средств вообще и семантического программирования в частности.

Мировая информационная среда демонстрирует впечатляющие темпы развития. Принципы, заложенные в ее основу, обладают огромным потенциалом. Создатели этих технологий не могли и представить, насколько глобальным окажется их детище. Сегодня наступает этап, когда требуются новые прорывные подходы, чтобы вовремя решить накопившиеся проблемы, пока эти проблемы не превратились в существенный тормоз. Главной проблемой на сегодняшний день является огромный объем неупорядоченной информации, хранящийся в Интернете. Природа человека не позволяет ему одновременно обрабатывать большое количество данных. Ему важно получать информационную «квинтэссенцию», дающую возможность принимать решения о дальнейшей углубленной работе в том или ином информационном блоке. Для того чтобы организовать такой последовательный процесс, информация должна быть ранжирована в некоторую иерархию, позволяющую пользователю шаг за шагом опускаться от менее подробной, но более общей информации, к информации детализированной, но локальной. Со своей стороны, компьютер способен предельно быстро «прочитывать» документы. Но у него другая трудность — он мало способен понимать человеческие тексты. В Интернете же огромный объем информации «складирован» в форматах, «непонятных» компьютеру, причем в совершенно хаотичном порядке. Когда на запрос поисковая машина выдает список из нескольких миллионов документов, это никого не удивляет. Объемы огромные, а система поиска, основанная на ключевых словах — примитивна.

Объясняется эта проблема довольно банальным образом: современные информационные сети в первую очередь настроены на хранение и распространение контента, ориентированного на понимание и обработку человеком, но никак не самим компьютером. Роль компьютеров сводится, как правило, к функциям хранения, транспортировки дан-

ных и их визуализации по указке человека. Анализ имеющейся информации для компьютера затруднен, поскольку структура этой информации совершенно к этому не подготовлена. Например, веб-страницы, имеющие разметку на языке HTML, являются для компьютера «черным ящиком», поскольку разметка подсказывает компьютеру, как «нарисовать» страничку, но несколько не способствует тому, чтобы компьютер взял на себя более «интеллектуальные» функции. Все это выпадает на долю человека, перегружая его работой, поскольку удел компьютера — работа с данными, а не знаниями.

Сегодня сформировалось довольно единодушное мнение о том, что в современной информационной среде существует тяжелый перекосяк в сторону внешнего вида информации по сравнению с описанием семантики электронных документов в доступной компьютеру форме. В частности, основная масса текстовых процессоров работает лишь с маленькой частью конструкций логического уровня. Они порождают то, что Сент-Лорейн в 1999 г. назвал WYSIWYG-кошмаром (WYSIWYG disaster). «Простой текст, пусть и слишком примитивный, обрабатывается намного легче, чем результат работы среднего редактора текстов, либо настольной издательской системы. На практике девиз WYSIWYG превращается в WYSIAYG — “What you see is ALL you get” («получишь только то, что видишь»). Текст настолько перегружен форматизирующими тегами, что совершенно не остается места для семантики и прагматики документа». «Форматирование является свойством текстов, которое позволяет им симпатично выглядеть, но совершенно не способно ответить на более фундаментальный вопрос о том, что они означают» [29].

## § 1. Онтологии и семантическое программирование

Автоматизированная работа с предметными областями начинается с определения строгого формализованного языка, ориентированного на описание этих предметных областей. Именно формальные описания служат источником «знаний» для компьютера, когда он манипулирует ресурсами. Ниже будет построена формальная система (точнее, семейство формальных систем), ориентированная на обработку информации в распределенных информационных средах и основанная на парадигме семантического программирования.

Основным объектом распределенных информационных сред является понятие *ресурса*. Наиболее общий и в то же время весьма продуктивный подход дает следующее определение: ресурс это то, чему дано *уникальное имя (денотат)*. Как только объект/сущность получает уникальное имя, он/она превращается в ресурс. Определенный таким образом ресурс отличается следующими качествами:

1. Благодаря уникальному имени, ресурс можно отличить от других ресурсов.
2. Благодаря уникальному имени, к ресурсу может быть привязано описание (мета-описание), позволяющее манипулировать информацией об этом ресурсе.
3. С помощью уникального имени часто можно получить доступ к самому ресурсу.

Например, третье свойство реализуется, когда таким именем служит адрес в Интернете. Конечно, хорошо, когда имеется метод, позволяющий по имени ресурса добраться до самого этого ресурса. Однако такая возможность не является обязательной.

Сегодня проблемы описания ресурсов и предметных областей в форматах, допускающих эффективную работу, находятся в фокусе внимания многих исследовательских групп, в частности консорциума W3C [25]. Подход W3C сформулирован в концептуальной работе [41]. В частности, в рамках консорциума были разработаны язык описания ресурсов RDF [31, 42–45] и язык описания онтологий OWL [33], стандарт которого был принят совсем недавно. В этом направлении действовала отдельная рабочая группа [24, 32]. Определенные усилия были потрачены на построение адекватных логических исчислений, лежащих в основе формализмов представления знаний в интернете [34, 35]. Эти формализмы в основном базируются на дескриптивных логиках [36]. Важнейшей составляющей общей проблемы является глобализация механизмов обмена знаниями, реализуемая через принятие открытых стандартов, через общий язык обмена данными XML [28–30], пространства имен [46], «всемирный алфавит» UNICODE [27], а также стандартный подход к генерированию уникальных имен ресурсов [26]. Очень большое значение имеют универсальные системы (мета)описаний ресурсов, которые позволяют на глобальном уровне обмениваться данными о ресурсах, осуществлять быстрый поиск информации, работать автоматизированным сервисам и т. д. В этом направлении также развивается ряд серьезных проектов, наиболее сбалансированным из которых является DublinCore [47–53]. Следует отметить и большое количество корпоративных стандартов, ориентированных на метаописание данных (например, в области образования [37–39]).

С содержательной точки зрения, информационная среда нового поколения базируется на формализованных системах представления знаний. Такие описания основаны на хорошо известных методах, в частности, классификациях (таксономиях), которые длительное время используются для описания обычных ресурсов (например, книг — достаточно вспомнить классификацию УДК [54]). Большое значение имеют таксономии и в виртуальной среде [40]. Но на сегодняшний день в качестве базовой модели представления знаний о предметных областях выбрана структура более общего характера — *онтология*. Предметная область в онтологиях имеет формат таксономии. Но, кроме этого, онтология включает в себя описание свойств предметной области и взаимодействия объектов на некотором формальном языке, имеющем логическую семантику. В онтологии объекты через иерархию классов и описание свойств связаны различными отношениями с другими объектами. Сами отношения также могут обладать полезными для получения информации свойствами (например, транзитивностью или симметричностью). Поэтому, как только некоторое значение из онтологии включается в метаописание некоторого ресурса, сам этот ресурс вовлекается в целую систему взаимосвязей, определяемую данной онтологией. Если в онтологии формализованы богатые знания, то это обеспечивает принципиально более качественные возможности для поисковых машин и других сервисов, использующих «интеллектуальную» обработку информации. Механизмы построения онтологий должны быть, очевидно, стандартизированы, что и сделано в рамках языка OWL [33]. «Интеллектуальная» обработка знаний базируется на системах вывода нужной информации из имеющихся знаний. Именно на базе этой системы вывода может быть построен нетривиальный поиск по информационной среде, сформированы продвинутые сервисы, манипулирующие интернет-ресурсами. Специфика огромного и

децентрализованного информационного пространства Интернета определяет ряд особенностей данной логической системы:

1. Устойчивость к «логически агрессивной» информационной среде: нет гарантии непротиворечивости и корректности опубликованных знаний.
2. Опыт показывает, что в силу огромных размеров интернета даже слабые и заведомо неполные алгоритмы поиска информации дают хорошие результаты. Логический формализм должен быть алгоритмически разрешимым и иметь разумную сложность вычислений, чтобы на нем могли работать автоматические сервисы в реальном времени.
3. Логический формализм должен поддерживать иерархию классов объектов и механизмы множественного наследования.
4. Логический формализм должен эффективно работать с имеющимися в Интернете ресурсами «первого» поколения, поддерживая основные принципы обработки информации.

К сожалению, традиционные методы автоматического доказательства и системы обработки знаний (например, экспертные системы) оказываются малоприспособленными для решения поставленной задачи. Здесь нужны новые подходы. Цель данной работы — показать, что полезным формализмом для такой работы может оказаться некоторый диалект семантического программирования (который назовем для краткости теорией информационных ресурсов, ТИР). Для демонстрации этого построим эскиз данной теории. В рамки этой статьи не входит глубокое исследование свойств ТИР и ее сравнение с другими формализмами — это работа на будущее. Большинство идей только обозначено и почти не развивается. Мы определим базовые конструкции формальной системы, сформулируем некоторые основные свойства. Больше внимание будет уделено обоснованию естественности конструкций ТИР, их «близости к реальному миру», с одной стороны, и демонстрации универсальности — с другой.

Ядром ТИР является базовая модель типов данных  $\mathfrak{R} = \langle M_1, \dots, M_s; \Omega \rangle$ , на основе которой строятся все остальные конструкции. Чтобы иметь возможность работать с типами данных базовой модели  $\mathfrak{R} = \langle M_1, \dots, M_s; \Omega \rangle$ , заведем для них специальные имена:  $\mathbf{DT} = \{\mathbf{dt}_1, \dots, \mathbf{dt}_s\}$ , символизирующие основные множества  $\mathfrak{R}$ , определив  $I(\mathbf{dt}_i) = M_i$ , где  $I$  — интерпретация. ТИР базируется на положениях, кратко представленных выше. Если подытожить то, что было сказано, получается, что в системах метаописаний предметных областей в основном фигурируют три позиции:

1. Иерархия объектов предметной области (классы объектов и понятие наследования), которая реализуется через таксономии и характеризуется набором дополнительных свойств, представленных на некотором логическом языке.
2. Набор атрибутов объекта предметной области, индивидуальным образом характеризующий его и выделяющий среди других объектов.
3. Взаимодействие классов реализуется через привязку к классу набора атрибутов, характеризующих объект как представителя данного класса.

Например, если имеем иерархию классов **профессор**  $\succ$  **ппс**  $\succ$  **человек** (**ппс** расшифровывается как «профессорско-преподавательский состав»), то если объект при-

надлежит к классу «профессор», то этот объект как человек обладает фамилией, датой рождения и, возможно, супругой/супругом; как ППС он обладает атрибутом «вуз», в котором работает и, наконец, как профессор он характеризуется атрибутом «кафедра», на которой получил звание профессора.

Пусть дана некоторая предметная область  $D$ . Предположим, анализ показывает, что данную предметную область удобно описывать, пользуясь иерархией классов с именами  $\mathbf{cn}_i$ . Любой класс является подмножеством предметной области  $D$ , т. е.  $I(\mathbf{cn}_i) \subseteq D$ , где  $I(\mathbf{cn}_i)$  — множество объектов класса с именем  $\mathbf{cn}_i$ . Во всех практически значимых случаях нам достаточно иметь конечное множество классов, которое обозначим  $\mathbf{CN} = \{\mathbf{cn}_1, \mathbf{cn}_2, \dots, \mathbf{cn}_\rho\}$ ,  $\rho < \omega$ . Тогда наследование представляет собой частичный порядок, заданный на  $\mathbf{CN}$ : то, что  $\mathbf{cn}_i$  наследует класс  $\mathbf{cn}_j$  (является его подклассом), будет обозначаться как  $\mathbf{cn}_i \succ_D \mathbf{cn}_j$ . Если характеризовать классы с информационной точки зрения, то об элементах наследующего (под)класса мы имеем больше (точнее, не меньше) информации, чем об элементах наследуемого (над)класса. Это означает, что  $\mathbf{cn}_i \succ_D \mathbf{cn}_j$  влечет  $I(\mathbf{cn}_i) \subseteq I(\mathbf{cn}_j)$ , т. е., чем больше информации о классе, тем меньше сам класс. Например, поскольку профессор является человеком, то о профессоре имеем информацию как о человеке, плюс дополнительную информацию, характеризующую его как профессора.

Очевидно, что атрибуты, имена которых будем обозначать через  $\mathbf{p}_i$ , должны привязываться к классам. Например, атрибут «фамилия» присущ классу людей, но не имеет смысла для класса автомобилей. С другой стороны, аргумент (значение атрибута) также должен быть либо элементом некоторого класса, либо элементом базовой модели данных  $\mathfrak{X} = \langle M_1, \dots, M_s; \Omega \rangle$ , т. е. значения атрибутов также должны быть типизированы. Отсюда появляется еще одно отношение, привязывающее атрибуты к классам. Выражение  $\mathbf{p}_i <_D \langle \mathbf{cn}, \mathbf{m} \rangle$ , где  $\mathbf{cn} \in \mathbf{CN}$  и  $\mathbf{m} \in \mathbf{CN} \cup \mathbf{DT}$ , означает, что в предметной области  $D$  атрибут с именем  $\mathbf{p}_i$  характеризует элементы класса  $\mathbf{cn}_j$ , а значениями атрибута, в зависимости от ситуации, могут быть элементы класса или основного множества  $\mathbf{m}$ . Например, в предметной области общественных связей появляется отношение **фамилия**  $<_D$  **человек, строка**. Здесь значением атрибута является строка, т. е. элемент базовой модели. В случае **супруг**  $<_D$  **человек, строка** значение атрибута выбирается уже не из базовой модели, а берется из класса «человек». В практически значимых случаях система описаний каждой конкретной предметной области использует, как правило, лишь конечное число атрибутов. Множества атрибутов будем обозначать через  $\mathbf{Attr} = \{\mathbf{p}_1, \dots, \mathbf{p}_k\}$ .

Наконец, в соответствии с концепцией ресурсов, нам нужно иметь возможность именовать объекты предметной области. Далеко не все объекты предметной области обязаны обладать именем. Возможны и безымянные объекты. Опять же, на практике нам имеет смысл использовать только конечные множества имен для каждой отдельной предметной области. С одной стороны, имена в нашей системе моделируют практические ситуации, когда таких имен требуется конечное количество. С другой стороны, конечность множества имен позволяет их использовать в более выразительном и гибком контексте. Множество имен, ориентированных на предметную область  $D$ , обозначим

через  $\mathbf{ID} = \{\mathbf{id}_1, \dots, \mathbf{id}_q\}$ .

Таким образом, общая понятийная структура предметной области  $D$  (ее «каркас») определяется набором  $\langle \mathbf{DT}, \mathbf{CN}, \mathbf{Attr}, \mathbf{ID}, \succ_D, <_D \rangle$ . Эту шестерку будем называть *онтологической формой* (*онтоформой*) предметной области  $D$ :

**Определение.** Пусть  $D$  — предметная область, а  $\mathbf{CN}, \mathbf{Attr}, \mathbf{ID}$  — конечные множества имен классов, атрибутов и объектов, определенных для описания предметной области  $D$ . Пусть также  $\succ_D$  — частичный порядок определенный на множестве  $\mathbf{CN}$ , а отношение  $<_D$  такое, что для каждого  $\mathbf{p}_i \in \mathbf{Attr}$  оно выполняется ровно на одной паре  $\mathbf{p}_i <_D \langle \mathbf{cn}, \mathbf{m} \rangle$ , где  $\mathbf{cn} \in \mathbf{CN}$  и  $\mathbf{m} \in \mathbf{CN} \cup \mathbf{DT}$ . Тогда онтологической формой (онтоформой) предметной области  $D$  назовем набор

$$\mathbf{K}_D = \langle \mathbf{DT}, \mathbf{CN}, \mathbf{Attr}, \mathbf{ID}, \succ_D, <_D \rangle.$$

Дальнейшие наши действия будут следующими. Над базовой моделью типов данных  $\mathfrak{R} = \langle M_1, \dots, M_s; \Omega \rangle$  построим наследственно-конечную надстройку  $T_{\mathfrak{R}}^{Attr}$  специального вида, которую будем называть *термальной надстройкой*. Грубо говоря, термы будут конструироваться из «функциональных аналогов» множеств  $\mathbf{CN}, \mathbf{Attr}$  и  $\mathbf{ID}$ . Поскольку предназначение  $T_{\mathfrak{R}}^{Attr}$  — служить источником для формальных описаний самых разных и любых предметных областей, то для построения  $T_{\mathfrak{R}}^{Attr}$  будем использовать бесконечную сигнатуру, чтобы иметь достаточно «материала» для описания сколь угодно большого количества предметных областей. Элементы этой надстройки (термы) будут служить описаниями (аппроксимациями) элементов предметной области, причем между элементами  $T_{\mathfrak{R}}^{Attr}$  также естественным образом определяется отношение аппроксимации: для  $t_1, t_2 \in T_{\mathfrak{R}}^{Attr}$ ,  $t_1$  аппроксимирует  $t_2$  (обозначается  $t_1 \sqsubseteq t_2$ ), если  $t_2$  содержит «не меньше» информации, чем  $t_1$ . Классы определим как подмножества  $T_{\mathfrak{R}}^{Attr}$ , обладающие специальными свойствами. Классы из  $T_{\mathfrak{R}}^{Attr}$  будут аппроксимировать содержательные классы предметной области  $D$ .

В наших построениях учтем также базовое понимание ресурса как объекта/сущности, обладающей уникальным именем. Будет задана некоторая система именования, привязывающая имена/идентификаторы к объектам надстройки  $T_{\mathfrak{R}}^{Attr}$ .

## § 2. Термальная надстройка над моделью: описание объектов

Определим термальную надстройку над базовой моделью типов. Формировать надстройку будем из синтаксического материала — термов специального вида. Для этого определим язык термов  $L_t$ , включающий константы, одноместные и двуместные функциональные символы. Будем считать все элементы модели  $\mathfrak{R} = \langle M_1, \dots, M_s; \Omega \rangle$  выделенными. Обозначим множество всех констант, соответствующих элементам  $\mathfrak{R}$ , через  $\bar{M} = \bar{M}_1 \cup \dots \cup \bar{M}_s$ . Составляющие языка  $L_t$  следующие.

1. Множество констант  $\bar{M}$ .
2. Константы  $\perp$  и  $\top$ .
3. Счетное множество констант  $ID = \{id_1, id_2, \dots\}$ , которые будут использоваться как имена объектов/ресурсов.

4. Счетное множество одноместных функциональных символов  $Attr = \{p_1, p_2, \dots\}$ , которые будем называть *атрибутами объектов*.
5. Счетное множество двуместных функциональных символов  $CN = \{cn, cn_2, \dots\}$ , которые будут ответственны за формирование описаний объектов как представителей соответствующих классов.
6. Конструктор конечных множеств  $\{\dots\}$ . Если  $t_1, \dots, t_q$  — термы  $L_t$ , то  $\{t_1, \dots, t_q\}$  — также терм  $L_t$ . Терм пустого множества будем обозначать константой  $\bar{\emptyset}$  (с крышкой — чтобы не путать с самим пустым множеством  $\emptyset$ ). Чтобы подчеркнуть синтаксическую природу этих объектов, будем называть их *термальными множествами*.

Язык  $L_t$  бесконечен и может использоваться для построения описаний предметных областей в соответствии с самыми разнообразными онтоформами, каждая из которых, по определению, обладает конечной структурой. В дальнейшем конечное подмножество  $L_t$ , соответствующее онтоформе  $\mathbf{K}_D$ , будем обозначать через  $L_t^D$ .

Построим наследственно-конечную надстройку над моделью  $\mathfrak{R}$  с использованием атрибутов. Будем называть ее *термальной*, поскольку элементами этой надстройки являются термы языка  $L_t$  — но не любые термы, а специальным образом сконструированные. Пусть  $T$  — некоторое множество термов языка  $L_t$ . Обозначим через  $Lin(T)$  совокупность всевозможных термальных множеств языка  $L_t$  вида  $\{t_1, \dots, t_k\}$ ,  $t_i \in T$  (т. е. построенных на основании п. 5 определения  $L_t$ ). Обозначим также

$$Attr(T) = \{p(t) \mid p \in Attr \ \& \ t \in T\}.$$

Наконец, нам нужно еще одно обозначение. Пусть  $t$  — терм. Обозначим через  $CN^t$  множество функциональных символов  $cn \in CN$ , входящих в терм  $t$ . Определим термальную надстройку как наименьшую неподвижную точку некоторого монотонного оператора на множествах термов языка  $L_t$ .

**Определение.** Термальная надстройка над  $\mathfrak{R}$  относительно набора атрибутов  $Attr$  (обозначается  $T_{\mathfrak{R}}^{Attr}$ ):

1.  $T_{\mathfrak{R},0}^{Attr} = \emptyset$ .
2. Пусть  $Set_n = Lin(Attr(\bar{M} \cup ID \cup T_{\mathfrak{R},n}^{Attr} \cup \{\perp\}))$ . Тогда

$$T_{\mathfrak{R},n+1}^{Attr} = T_{\mathfrak{R},n}^{Attr} \cup \{cn(c, a) \mid cn \in CN \setminus CN^t, c \in Lin(T_{\mathfrak{R},n}^{Attr}), a \in Set_n\}.$$

3.  $T_{\mathfrak{R}}^{Attr} = \bigcup_n T_{\mathfrak{R},n}^{Attr} \cup \{\top\}$ .

Таким образом,  $T_{\mathfrak{R}}^{Attr}$  состоит из термов вида  $cn(c, a)$ , где  $c = \{t_1, \dots, t_q\}$ ,  $t_i \in T_{\mathfrak{R}}^{Attr}$ ,  $a$  — множество термов вида  $p(t)$ , где либо  $t \in T_{\mathfrak{R}}^{Attr}$ , либо  $t \in \bar{M}$ , либо  $t = \perp$ . Специальная константа  $\perp$  неформально означает «неопределенное» значение. Условие  $cn \in CN \setminus CN^t$  во втором пункте определения означает, что каждый функциональный символ  $cn \in CN$  может входит в отдельный терм не более одного раза. Кроме того, в  $T_{\mathfrak{R}}^{Attr}$  отдельно входит константа  $\top$ . В дальнейшем будем пользоваться обозначением  $T_{\mathfrak{R}\mathbf{E}}^{Attr} = T_{\mathfrak{R}}^{Attr} \setminus \{\top\}$ .



**Определение.** Именованиием назовем произвольное инъективное отображение  $I$  из множества  $ID$  на подмножество  $T_{\mathfrak{R}}^{Attr}$ , т. е.  $I : ID \rightarrow T_{\mathfrak{R}}^{Attr}$ .

Если  $I(id) = t$ , то  $id$  назовем *именем (идентификатором)* элемента  $t$ . Элементы  $T_{\mathfrak{R}}^{Attr}$ , не имеющие имени, назовем *анонимными*.

**Пример.** Описание двух объектов:

$$I(id_{\text{петров}}) = \text{студент}(\{\text{человек}(\{\text{объект}(\bar{\emptyset}, \bar{\emptyset}), \\ \{\text{фамилия}(\text{"Петров"}), \text{супруг}(id_{\text{петрова}})\})\}, \{\text{вуз}(id_{\text{нгу}})\})$$

$$I(id_{\text{петрова}}) = \text{человек}(\{\text{объект}(\bar{\emptyset}, \bar{\emptyset}), \{\text{фамилия}(\text{"Петрова"}), \text{супруг}(id_{\text{петров}})\})\})$$

содержит информацию о двух супругах, один из которых учится в вузе с именем  $id_{\text{нгу}}$ . Здесь функциональные символы  $\text{студент}$ ,  $\text{человек} \in CN$ ,  $\text{фамилия}$ ,  $\text{супруг}$ ,  $\text{вуз} \in Attr$ ,  $id_{\text{петров}}$ ,  $id_{\text{петрова}}$ ,  $id_{\text{нгу}} \in ID$ .

Из примеров видно, что подтермы элементов  $T_{\mathfrak{R}}^{Attr}$  играют разную роль. Например, если подтерм является аргументом некоторого атрибута  $p$ , то он описывает не текущий объект, а другой объект, который «привязан» к данному объекту через атрибут некоторой смысловой связью — как  $id_{\text{петрова}}$  в атрибуте  $\text{супруг}(id_{\text{петрова}})$ . Терм  $id_{\text{петрова}}$  относится не к самому объекту, а другому объекту, который связывается с данным через атрибут. Таким образом, подтермы данного терма можно разбить на уровни в зависимости от того, какое количество атрибутов является их надтермами. Только *непосредственные подтермы* (подтермы нулевого уровня, не входящие в атрибуты) содержат информацию, непосредственно относящуюся к данному объекту. Все остальные описывают какие-то другие объекты, связанные с данным.

**Определение.** Терм  $t_1$  является непосредственным подтермом терма  $t$  если не существует подтерма  $t$  вида  $p(t_2)$  такого, что  $t_1$  является подтермом  $p(t_2)$ .

### § 3. Иерархия и наследование информации

Термальная надстройка при соответствующей интерпретации вполне корректно реализует концептуальную схему, кратко описанную в начале данной работы. Вернемся к примеру с представителями высшего образования, каждый из которых может принадлежать сразу нескольким классам «человек», «профессор», «пшс», «студент», «бухгалтер» и т. д. Как уже отмечалось выше, с теоретико-множественной точки зрения наследование реализуется через включение одного множества в другое:  $I(\text{профессор}) \subseteq I(\text{ппс}) \subseteq I(\text{человек})$ . Как правило, такой порядок по включению является частичным, и некоторые классы не сравнимы. Допускается множественное наследование: например, объект может принадлежать одновременно несравнимым классам «студент» и «родитель». Каждому классу соответствуют определенные атрибуты (например, атрибутом «родителя» является его ребенок). Кроме того, если класс  $\mathbf{cn}_1$  наследует от класса  $\mathbf{cn}_2$ , то атрибуты  $\mathbf{cn}_2$  также являются и атрибутами  $\mathbf{cn}_1$ . Удобно, когда все классы объектов, являются наследниками самого общего класса с именем **объект**, что может быть учтено через включение терма  $\text{объект}(\bar{\emptyset}, \bar{\emptyset})$  с пустым набором информации.

Элементы надстройки  $T_{\mathfrak{R}}^{Attr}$  над моделью  $\mathfrak{R}$  с этой точки зрения интерпретируются как описания объектов. Двуместные функциональные символы  $cn \in CN$  объединяют разрозненную информацию об объекте в рамках одного терма. В дальнейшем терм  $cn(c, a) \in T_{\mathfrak{R}}^{Attr}$  будет интерпретироваться как описание некоторого объекта, принадлежащего классу с именем  $cn$ . При этом первый аргумент  $cn$  определяет данный элемент как наследника элементов других более общих классов с соответствующими атрибутами. Второй аргумент содержит множество атрибутов объекта, присущих «текущему» классу  $cn$ . При этом модель  $\mathfrak{R}$  определяет базовые типы данных, на основе которых строятся описания объектов. Обратим внимание на принципиальную важность именования. Например, для анонимных элементов невозможно определить понятие «супруг», поскольку в силу симметричности этого понятия получится терм с циклом, т. е. бесконечной глубины. Присутствие в элементе атрибута вида  $p(\perp)$  означает, что для данного элемента атрибут  $p$  не имеет никакого значения. Это не эквивалентно отсутствию атрибута в терме. Отсутствие атрибута означает, что *неизвестно*, какое значение имеет данный атрибут для данного объекта, в то время, как присутствие  $p(\perp)$  можно интерпретировать так: «*известно*, что использование атрибута  $p$  для описания данного объекта не имеет смысла».

Ниже будет показано, что определенная выше термальная надстройка над моделью типов данных представляет собой универсальную схему, позволяющую в едином стиле моделировать разнообразные форматы представления знаний в Интернете и не только. Но прежде определим формальный эквивалент только что приведенных неформальных рассуждений.

Чтобы работать с информацией, находящейся в «глубине» термов из надстройки  $T_{\mathfrak{R}}^{Attr}$ , введем два бинарных отношения — отношение наследования  $t_1 \succ t_2$ , определенное на элементах  $T_{\mathfrak{R}}^{Attr}$ , и отношение принадлежности атрибута элементу  $p(t_1) < t_2$ . Заметим, что выше мы уже использовали похожие символы отношений  $\succ_D$  и  $<_D$ , определенных на именах классов и атрибутов из множеств  $\mathbf{P}$  и  $\mathbf{CN}$ . Между  $\succ$  и  $\succ_D$ ,  $<$  и  $<_D$  есть прямая связь, которая будет установлена нами ниже.

Структура элементов  $T_{\mathfrak{R}}^{Attr}$  устроена таким образом, что атрибуты в них не сгруппированы в одном месте, а «распределены» по классам. Принципы распределения атрибутов по классам и их взаимодействие (в голове человека) достойны серьезного анализа. Однако это выходит за рамки настоящей публикации. Заметим только, что в реальной жизни термины совершают «прыжки» из одного класса в другой. Например, атрибут «фамилия» привязывается к классу «человек». Однако, при официальном обращении к одному из авторов этой публикации, его, как правило, называют «профессор Манцивода», а не «человек Манцивода». Это вполне корректно, поскольку, появившись в рамках некоторого класса атрибут «проецируется» на все классы, которые наследуют информацию данного класса. Отсюда появляется отношение «наследования», работающее на элементах  $T_{\mathfrak{R}}^{Attr}$ .

**Определение.** Пусть  $t_1, t_2 \in T_{\mathfrak{R}}^{Attr}$ . Будем говорить что элемент  $t_1 = cn(c, a)$  является *непосредственным наследником* элемента  $t_2$  (обозначается  $t_1 \succ^* t_2$ ), если  $t_2 \in c$  (т. е. терм  $c$  имеет вид  $\{\dots, t_2, \dots\}$ ). Элемент  $t_1$  является *наследником*  $t_2$  (обозначается  $t_1 \succ t_2$ ), если либо  $t_1 \succ^* t_2$ , либо имеется непосредственный наследник  $t_3 \in c$  такой, что  $t_3 \succ t_2$ .

Обратим внимание, что если  $t_1 \succ t_2$ , то  $t_2$  является подтермом первого аргумента  $t_1$  (точнее, непосредственным подтермом). Заметим также, что с неформальной точки зрения  $t_1 \succ t_2$  означает что  $t_1$  и  $t_2$  описывают один и тот же объект. Однако  $t_1$  описывает его как объект класса, являющегося наследником класса из  $t_2$ . В частности, объект «Петров» как студент является наследника объекта «Петров» как человека.

Теперь определим рекурсивное отношение  $<$  принадлежности атрибута элементу. Отношение  $p(t') < t$  действует следующим образом.

**Определение.** Пусть  $t_1 = cn(c, a)$ . Будем говорить, что атрибут  $p(t_2)$  *непосредственно принадлежит* элементу  $t_1$  (обозначается  $p(t_2) <^* t_1$ ), если  $p(t_2) \in a$ . Будем говорить, что атрибут  $p(t_2)$  *принадлежит* элементу  $t_1 = cn(c, a)$  (обозначается  $p(t_2) < t_1$ ), если либо  $p(t_2) <^* t_1$ , либо существует  $t_3$  такой, что  $t_1 \succ t_3$  и  $p(t_2) <^* t_3$ .

Неформально, атрибут  $p(t')$ , где  $t' \neq \perp$ , принадлежит терму  $t$ , если он является характеристикой некоторого класса, которому принадлежит объект предметной области, описываемый термом  $t$ . Непосредственная принадлежность означает, что атрибут является характеристикой «внешнего» класса **cn**. Если же  $p(\perp) < t$ , то это означает, что атрибут  $p$  не может быть характеристикой  $t$ . Легко показать также, что если  $p(t') < t$ , то  $p(t')$  является непосредственным подтермом  $t$  (что согласуется с тем, что  $p(t')$  характеризует именно сам описываемый в  $t$  объект, а не какие-то другие объекты).

Основная содержательная информация об объектах предметной области как отдельных индивидуумах хранится в атрибутах. С этой точки зрения, распределение атрибутов по классам больше имеет роль для понимания того, как тот или иной элемент был построен, т. е. с интенциональной точки зрения. С точки зрения экстенциональной важно то, какие атрибуты в элементе имеются и какие значения они принимают. С экстенциональной точки зрения атрибут принадлежит объекту не как представителю какого-то класса, а как некоторому отдельному индивидууму. Такое понимание служит основой для «экстенциональной» (объемной) интерпретации информации, заложенной в элементы множества термов  $T_{\mathfrak{R}}^{Attr}$ , которое реализуется через следующее определение.

**Определение.** Пусть  $t \in T_{\mathfrak{R}}^{Attr}$ . Обозначим через  $ext(t)$  множество всех атрибутов  $p_i(t_i) < t$ , т. е.  $ext(t) = \{p_i(t_i) \mid p_i(t_i) < t\}$ . Множество  $ext(t)$  назовем экстенционалом элемента  $t$ .

**Замечание.** Атомарная формула  $p(t_1) < t_2$  понимается нами как высказывание «объект  $t_2$  обладает свойством  $p$  со значением  $t_1$ ».

Это немного отличается от стандартного для логики первого порядка подхода, когда свойство определяется в форме двуместного предиката, представляемого атомарной формулой  $p(t_1, t_2)$ . Объясняется это тем, что стандартный подход методологически не очень нам подходит, поскольку тогда получается, что свойства строятся над множеством объектов «сверху» и как бы отдельно от них. Для нас же существенно, чтобы свойство объекта было явным образом к нему привязано и, если можно так сказать, содержалось «внутри него». Это отличает объектный подход, в котором объекты строятся явно, от аксиоматического подхода, когда объекты строятся опосредованно — как сущности, удовлетворяющие определенным законам. Явная привязка свойств к объектам позво-

ляет довольно естественным образом определить важнейшие понятия наследования и аппроксимации. Это влияет и на практическую реализацию данной схемы, поскольку позволяет группировать информацию об объекте в одном месте, а не распределять по отдельным предикатам. Другими словами, данная схема позволяет естественным образом объединить логический подход с очень успешным на сегодняшний день объектно-ориентированным подходом, что может оказаться важным для перспектив развития наших конструкций. Поэтому мы и моделируем свойства объектов не с помощью предикатов, а в виде термов специального вида.

Рассмотрим теперь вопрос эквивалентности элементов  $t \in T_{\mathfrak{R}}^{Attr}$ . Поскольку эти элементы описывают объекты предметной области, то существенным является тот объем информации, который заложен в каждый отдельный элемент и неважно, в каком порядке эта информация в терме представлена. С другой стороны, поскольку элементы  $T_{\mathfrak{R}}^{Attr}$  являются синтаксическими объектами, вполне допустимы термы вида  $\{t_1, t_1\}$ , эквивалентные  $\{t_1\}$ , графически отличными являются термы  $\{t_2, t_1\}$  и  $\{t_1, t_2\}$ , содержащие равный объем информации и т.д.. Нам нет смысла вводить специальные аксиомы для множеств, поскольку с точки зрения отношений, работающих с информацией, например,  $f$  и  $<$ , эти термы и так будут неотличимы друг от друга. Такие неотличимые термы будем называть эквивалентными (обозначается  $t_1 \equiv t_2$ ).

#### Определение.

1. Элементы  $\bar{M} \cup \{\top, \perp\}$  эквивалентны только самим себе, т.е.  $t \equiv r$  тогда и только тогда, когда  $t = r$  (графическое совпадение).
2. Термальные множества  $s_1$  и  $s_2$  эквивалентны тогда и только тогда, когда для каждого  $t_1 \in s_1$  существует  $t_2 \in s_2$ , что  $t_1 \equiv t_2$  и для каждого  $t_2 \in s_2$  существует  $t_1 \in s_1$ , что  $t_1 \equiv t_2$ .
3.  $p_1(t_1) \equiv p_2(t_2)$  тогда и только тогда, когда  $p_1 = p_2$  и  $t_1 \equiv t_2$ .
4. Термы  $cn_1(c_1, a_1)$  и  $cn_2(c_2, a_2)$  эквивалентны тогда и только тогда, когда  $cn_1 = cn_2$ ,  $c_1 \equiv c_2$  и  $a_1 \equiv a_2$ .

Очевидно, что если  $t \equiv r$ , то  $ext(t) = ext(r)$ . Обратное верно далеко не всегда. Ниже будет показано что для «хорошо устроенных» иерархий классов верно и обратное утверждение. Но прежде сформулируем строгое определение понятий аппроксимации.

#### § 4. Аппроксимация и объединение информации

Отношение аппроксимации на элементах термальной надстройки (обозначается  $t_0 \sqsubseteq t_1$ ) позволяет сравнивать объем информации, хранящийся в термах. Отношение  $t_0 \sqsubseteq t_1$  истинно, если в элементе  $t_0$  хранится меньше (точнее, не больше) информации, чем в  $t_1$ . Аппроксимация определяется взаиморекурсивно совместно с понятием непосредственной аппроксимации.

**Определение.** Пусть  $cn_0, cn_1 \in CN$ . Элемент  $t_0 = cn_0(\dots)$  непосредственно аппроксимирует элемент  $t_1 = cn_1(\dots)$  (обозначается  $t_0 \sqsubseteq^* t_1$ ), тогда и только тогда, когда  $cn_0 = cn_1$  и:

1. Для каждого  $p(t) <^* t_0$ , существует  $p(t') <^* t_1$  такой, что  $t \sqsubseteq t'$ .
2. Для каждого  $t, t_0 \succ^* t$ , существует  $t'$  такой, что  $t_0 \succ^* t'$  и  $t \sqsubseteq t'$ .

**Определение (аппроксимация).** Для каждого  $id \in ID$  и  $t, t_0, t_1 \in T_{\mathfrak{R}}^{Attr}$ :

1.  $t \sqsubseteq id$  ( $id \sqsubseteq t$ ) тогда и только тогда, когда выполняется  $t \sqsubseteq I(id)$  ( $I(id) \sqsubseteq t$ ).
2.  $e \sqsubseteq T$  для любого  $e \in T_{\mathfrak{R}}^{Attr} \cup \bar{M} \cup \{\perp\}$ .
3. Элементы основных множеств модели  $\mathfrak{R}$  и константа  $\perp$  не сравнимы ни с какими элементами, кроме самих себя и  $\top$ , т. е.  
 $(\forall c \in \bar{M} \cup \{\perp\}) (\forall e \in T_{\mathfrak{R} \setminus \top}^{Attr} \cup \bar{M} \cup \{\perp\}) (e \sqsubseteq c \vee c \sqsubseteq e \rightarrow e = c)$
4.  $t_0 \sqsubseteq t_1$  тогда и только тогда, когда либо  $t_0 \sqsubseteq^* t_1$ , либо  $t_0 \sqsubseteq t'$  для некоторого  $t'$ ,  $t_1 \succ^* t'$ .

Обратим внимание, что отношение аппроксимации определено также на элементах основных множества модели  $\mathfrak{R}$ , несмотря на то, что никакие элементы основных множеств не являются элементами  $T_{\mathfrak{R}}^{Attr}$ . Элемент  $\top$  трактуется как наибольший (противоречивый, переопределенный) элемент. Элемент  $\perp$  в отличие от  $\top$  трактуется не как «наименьший», а как «неопределенный» элемент.

**Предложение.**

- (i) *Отношение аппроксимации рефлексивно и транзитивно.*
- (ii)  $t_1 \sqsubseteq t_2$  и  $t_2 \sqsubseteq t_1$ , тогда и только тогда, когда  $t_1 \equiv t_2$ .

Таким образом, пара  $(T_{\mathfrak{R}}^{Attr}, \sqsubseteq)$  образует частичный порядок. Определим операцию взятия наименьшей верхней грани элементов  $t_1 \sqcup t_2$ , служащей формальным аналогом операции объединения информации из двух разных описаний.

**Определение.** Пусть  $t, t_1, t_2 \in T_{\mathfrak{R}}^{Attr}$ . Элемент  $t$  назовем наименьшей верхней гранью (объединением) элементов  $t_1, t_2$  (обозначается  $t = t_1 \sqcup t_2$ ), если  $t_1 \sqsubseteq t$ ,  $t_2 \sqsubseteq t$  и для любого элемента  $t_0$  истинность  $t_1 \sqsubseteq t_0 \wedge t_2 \sqsubseteq t_0$  влечет  $t \sqsubseteq t_0$

Далеко не для любой пары термов  $t_1, t_2 \in T_{\mathfrak{R}}^{Attr}$  существует объединение. Например, термы человек( $\bar{\emptyset}, \bar{\emptyset}$ ) и профессор( $\bar{\emptyset}, \bar{\emptyset}$ ) оба аппроксимируют два несравнимых элемента

$$\begin{aligned} t'_1 &= \text{человек}(\{\text{профессор}(\bar{\emptyset}, \bar{\emptyset})\}, \emptyset) \text{ и} \\ t'_1 &= \text{профессор}(\{\text{человек}(\bar{\emptyset}, \bar{\emptyset})\}, \emptyset). \end{aligned}$$

Отметим однако, что с точки зрения нашей неформальной семантики первый терм описывает объект в предметной области, в которой класс «человек» является наследником класса «профессор», в то время, как второй — в предметной области, в которой «профессор» является наследником «человека». Отсюда получаем, что при фиксированной онтоформе  $\mathbf{K}_D$  одновременное существование в одной предметной области объектов, описанных элементами  $t'_1$  и  $t'_2$  невозможно. Таким образом, далеко не все элементы  $T_{\mathfrak{R}}^{Attr}$  могут служить описанием для элементов предметной области  $D$ . Во-первых, в силу бесконечности языка  $T_{\mathfrak{R}}^{Attr}$  содержит термы с функциональными символами, не соответствующими никаким именам из множеств  $\mathbf{CN}$  и  $\mathbf{Attr}$ . Во-вторых, даже если присутствуют «правильные» функциональные символы, то они могут входить в терм в порядке, не соответствующем иерархии наследования, определенной в онтоформе, а атрибуты могут быть привязаны к «неправильным» классам. В связи с этим фиксация

онтологической формы  $\mathbf{K}_D$  ограничивает набор элементов  $T_{\mathfrak{R}}^{Attr}$ , которые допустимо использовать для описания предметной области. Подмножество  $T_{\mathfrak{R}}^{Attr}$ , содержащее такие элементы и только их, назовем *согласованным с  $\mathbf{K}_D$*  и обозначим через  $T_{\mathfrak{R}}^D$ .

Пусть  $D$  — некоторая предметная область, а

$$\mathbf{K}_D = \langle \mathbf{DT}, \mathbf{CN}, \mathbf{Attr}, \mathbf{ID}, \succ_D, \triangleleft_D \rangle$$

— ее онтоформа. Если  $p(t') \triangleleft t$ ,  $t \in T_{\mathfrak{R}}^{Attr}$ , то будем говорить, что  $p$  входит в  $t$  на уровне  $cn_i \in \mathbf{CN}$ , если  $t$  содержит непосредственный подтерм вида  $cn_i(\dots, \{\dots p(t') \dots\})$ , в котором  $p(t')$  входит в термальное множество из второго аргумента. Вхождение атрибута на уровне  $cn_i \in \mathbf{CN}$  является синтаксическим аналогом того, что описываемый объект принадлежит классу с именем  $\mathbf{cn}_i$ , а атрибут  $\mathbf{p}$  является характеристикой объектов класса  $\mathbf{cn}_i$ .

**Определение.** Элемент  $t \in T_{\mathfrak{R}}^{Attr}$  назовем *согласованным с онтоформой  $\mathbf{K}_D$* , если

- (1) для любых подтермов  $t$  вида  $t_1 = cn_1(\dots)$  и  $t_2 = cn_2(\dots)$  (включая сам терм  $t$ ) выполняется  $t_1 \succ t_2$  тогда и только тогда, когда  $\mathbf{cn}_1 \succ_D \mathbf{cn}_2$ .
- (2) если  $p(m) \triangleleft t$ ,  $m \in \bar{M}$ , и  $p$  входит в  $t$  на уровне  $cn$ , то  $\mathbf{p} \triangleleft_D \langle \mathbf{cn}, \mathbf{dt} \rangle$ , где  $\mathbf{cn} \in \mathbf{CN}$ , и  $\mathbf{dt} \in \mathbf{DT}$ ,  $m \in I(\mathbf{dt})$ .
- (3) если  $p(t') \triangleleft t$ ,  $t' = cn'(\dots)$ , и  $p$  входит в  $t$  на уровне  $cn$ , то  $\mathbf{p} \triangleleft_D \langle \mathbf{cn}, \mathbf{cn}^* \rangle$ , где  $\mathbf{cn}, \mathbf{cn}^* \in \mathbf{CN}$ , и  $t$  согласован с  $\mathbf{K}_D$ , причем  $cn^*$  и  $cn'$  сравнимы относительно порядка  $\succ_D$  (либо  $cn' \succ_D cn^*$ , либо  $cn^* \succ_D cn'$ , либо  $cn^* \succ cn'$ ).

Первый пункт данного определения обеспечивает в согласованных терминах корректность наследования классов относительно онтоформы  $\mathbf{K}_D$ . Второй пункт обеспечивает «типизацию» для значений атрибутов в том случае, когда эти значения являются элементами основной модели  $\mathfrak{R}$ , т. е. базовыми типами данных. Третий пункт фиксирует случай, когда значением атрибута является другой объект предметной области. Здесь ситуация несколько тоньше. В соответствии с отношением  $\mathbf{p} \triangleleft_D \langle \mathbf{cn}, \mathbf{cn}^* \rangle$  значение атрибута должно принадлежать классу  $\mathbf{cn}^*$ . Проблема в том, что аргументы атрибутов не обязательно содержат полную информацию об объекте, который связан с данным через атрибут. Например, формализация фразы «Петров женат» может быть сформулирована так: «существует объект, являющийся супругой Петрова»:

$$\text{человек}(\{\text{объект}(\bar{\emptyset}, \bar{\emptyset})\}, \{\text{фамилия}(\langle \text{Петров} \rangle), \text{супруг}(\text{объект}(\bar{\emptyset}, \bar{\emptyset}))\})$$

Но типизация атрибута **супруг**, очевидно, дает  $\text{супруг} \triangleleft_D \langle \text{человек}, \text{человек} \rangle$ . Терм  $\text{объект}(\bar{\emptyset}, \bar{\emptyset})$  не содержит данных о том, что супруга Петрова является человеком, что означает, что не вся информация о данном объекте имеется в наличии. И это вполне «законная» ситуация, которую необходимо учитывать. Важно только, чтобы уже имеющаяся информация не противоречила факту принадлежности описываемого объекта соответствующему классу — это и отражается в определении. В данном примере условия определения соблюдаются, поскольку человек, очевидно, является частным случаем «объекта вообще», т. е. **человек**  $\succ$  **объект**. Возможна и обратная ситуация, когда  $\mathbf{cn}' \succ_D \mathbf{cn}^*$ . Допустимость этого случая объясняется тем, что в качестве аргумента

атрибута может использоваться представитель некоторого подкласса класса  $\mathbf{cn}^*$ , например, при формализации утверждения «Петров женат на профессоре».

**Определение.** Подмножество  $T_{\mathfrak{R}}^D \subset T_{\mathfrak{R}}^{Attr}$  назовем *согласованным* с онтоформой  $\mathbf{K}_D$ , если оно состоит из всех согласованных с  $\mathbf{K}_D$  элементов и  $\top$ , и только них.

Понятие согласованности выделяет в термальной надстройке те элементы, формат которых корректно отражает иерархию классов предметной области, описанием которой мы занимаемся. Только элементы из  $T_{\mathfrak{R}}^D$  пригодны для описания объектов  $D$  с иерархией  $\mathbf{K}_D$ . Проверка согласованности элемента с порядком является разрешимой задачей, имеющей линейные алгоритмы относительно сложности терма.

**Теорема.** Пусть  $T_{\mathfrak{R}}^D$  — согласованное множество онтоформы  $\mathbf{K}_D$ . Тогда для любых  $t_1, t_2 \in T_{\mathfrak{R}}^D$  существует единственный элемент  $t \in T_{\mathfrak{R}}^D$  такой, что  $t_1 \sqsubseteq t$ ,  $t_2 \sqsubseteq t$  и для любого элемента  $t_0$ , выполняется  $t_1 \sqsubseteq t_0 \wedge t_2 \sqsubseteq t_0 \supset t \sqsubseteq t_0$ .

Таким образом, в рамках  $T_{\mathfrak{R}}^D$  (т. е. , как только фиксируется предметная область и иерархия классов в ней) любые два элемента  $t_1$  и  $t_2$  имеют наименьшую верхнюю грань  $t_1 \sqcup t_2$ . Это утверждение весьма существенно для дальнейшего изложения. С неформальной точки зрения это означает, что если в конкретной предметной области  $D$  у нас есть две «информации»  $t_1$  и  $t_2$ , описывающие один и тот же объект  $d \in D$ , то мы можем сгруппировать их в рамках  $T_{\mathfrak{R}}^D$  в одну, которая в точности суммирует те знания о  $d$ , которые мы имеем по отдельности в  $t_1$  и  $t_2$ . С другой стороны, если  $t_1 \sqcup t_2 = \top$ , то это означает, что  $t_1$  и  $t_2$  заведомо не могут быть описаниями одного и того же объекта. Если получаем для одного и того же объекта два несовместных описания, то система описаний предметной области противоречива. Можно указать довольно простой (линейный относительно суммарной сложности термов) алгоритм нахождения наименьшей верхней грани двух термов.

Отметим еще одно полезное качество множества  $T_{\mathfrak{R}}^D$ :

**Предложение.** Для любых  $t_1, t_2 \in T_{\mathfrak{R}}^D$ , если  $ext(t_1) = ext(t_2)$ , то  $t_1 \equiv t_2$ .

Таким образом, в рамках  $T_{\mathfrak{R}}^D$  обеспечивается взаимосвязь между интенциональным и экстенциональным определением.

## § 5. Классы

Теперь имеются все средства для определения понятия класса в нашей системе.

**Определение.** Элемент  $t \in T_{\mathfrak{R}}^D$  назовем полным относительно онтоформы  $\mathbf{K}_D = \langle \mathbf{DT}, \mathbf{CN}, \mathbf{Attr}, \mathbf{ID}, \succ_D, \triangleleft_D \rangle$ , если для любого элемента  $t_0 \in T_{\mathfrak{R}}^D$  выполняется либо  $t \sqcup t_0 = t$ , либо  $t \sqcup t_0 = \top$ .

Таким образом, полные элементы содержат всю возможную в рамках онтоформы  $\mathbf{K}_D$  информацию и не могут быть расширены. Именованное  $I$  назовем *корректным*, если оно отображается только в полные элементы  $T_{\mathfrak{R}}^D$ . С содержательной точки зрения это означает, что корректные именованные имеют только максимально определенные элементы  $T_{\mathfrak{R}}^D$ . Требование корректности является весьма естественным, поскольку ос-

новная цель использования имен — идентифицировать объекты из предметной области. Но поскольку в контексте онтологической формы  $\mathbf{K}_D$  говорить о самих объектах мы можем только на языке  $\mathbf{L}_1^D$ , максимально что можно сделать для описания того или иного объекта, это определить его с помощью некоторого полного элемента  $T_{\mathfrak{R}}^D$ . Если же именовать неполный объект, то существует опасность того, что даже в рамках языка он может быть превращен в несколько разных полных элементов, что способно привести к появлению элементов-тезок. Нам же надо, чтобы имя уникальным образом идентифицировало объект (что позволит работать с объектами как ресурсами). В дальнейшем будем работать только с корректными именованями. Следующее достаточно очевидное утверждение дает существенную характеристику множества  $T_{\mathfrak{R}}^D$ :

**Предложение.** Для любого  $t \in T_{\mathfrak{R}}^D$  существуют полные элементы  $\bar{t} \in T_{\mathfrak{R}}^D$  такие, что  $t \sqsubseteq \bar{t}$ , причем любая строго возрастающая цепочка  $t \sqsubseteq t_1 \sqsubseteq \dots \sqsubseteq t_k \sqsubseteq \bar{t}$  является конечной.

Обозначим через  $\bar{T}_{\mathfrak{R}}^D$  множество всех полных элементов, входящих в  $T_{\mathfrak{R}}^D$ , а через  $\bar{T}_{\mathfrak{R}i}^D$  множество всех термов  $t \in \bar{T}_{\mathfrak{R}}^D$  таких, что  $cn_i(\bar{\emptyset}, \bar{\emptyset}) \sqsubseteq t$ .

**Определение.** Пусть  $\mathbf{K}_1^D$  — онтоформа. Допустимым классом с именем  $\mathbf{cn}_i$  является любое подмножество  $C_i \subseteq \bar{T}_{\mathfrak{R}i}^D$ .

Другими словами, при построении конкретной иерархии классов в качестве класса с именем  $\mathbf{cn}_i$  можно выбирать любое подмножество  $\bar{T}_{\mathfrak{R}i}^D$ . Таким образом, необходимое условие принадлежности элемента  $T_{\mathfrak{R}}^{Attr}$  классу с именем  $\mathbf{cn}_i$  всегда синтаксически определимо, поскольку все элементы класса имеют непосредственный подтерм вида  $cn_i(c, a)$  — именно это означает отношение  $cn_i(\bar{\emptyset}, \bar{\emptyset}) \sqsubseteq t$ . С другой стороны, элемент, обладающий данным свойством, может и не принадлежать классу — в случае, когда нет объекта предметной области, к которому применимо описание в виде этого элемента. В дальнейшем расширим функцию именованя  $I$ , определив ее на именах из  $\mathbf{CN}$ :  $I(\mathbf{cn}_i) \subseteq \bar{T}_{\mathfrak{R}i}^D$  — класс из иерархии классов, соответствующих предметной области  $D$ , который именуется константой  $\mathbf{cn}_i$ .

## § 6. Ресурсы

В начале данной работы мы рассматривали понятие ресурса как ключевое понятие распределенной информационной среды. С концептуальной точки зрения ресурс определяется как сущность/объект, имеющий уникальное имя. Формализм, который строится здесь, ориентирован на автоматизированную обработку ресурсов. Ресурсы могут быть сколь угодно сложными по объему информации и поведению. Чтобы реализовать нетривиальные компьютерные сервисы, манипулирующие ресурсами, необходимо обеспечить компьютер знанием о ресурсах. Это реализуется через механизм метаданных. В нашей терминологии метаописания — это аппроксимации ресурсов: чем больше информации о ресурсе, тем ближе аппроксимация к оригиналу. Этот подход реализован в предыдущих пунктах. Для каждой предметной области  $D$  фиксация онтологической формы  $\mathbf{K}_D$  определяет иерархию описаний ресурсов  $T_{\mathfrak{R}}^D$  с точностью до языка  $L_t^D$ , который, в свою



очередь, определяется моделью  $\mathfrak{R}$ , и онтоформой  $\mathbf{K}_D$ . Полные элементы в иерархии  $T_{\mathfrak{R}}^D$  — это тот максимум в описании объектов, который может быть достигнут в рамках данного языка.

Механизм именованя позволяет определять работу с ресурсами в соответствии с их концептуальным определением. Чтобы иметь явную возможность работать с ресурсами, введем понятие описания ресурса:

**Определение.** Описанием ресурса назовем формулу  $id :: t$ , где  $::$  — новый бинарный предикатный символ,  $id$  — имя ресурса, а  $t \in T_{\mathfrak{R}}^D$ . Описание назовем корректным относительно функции именованя  $I$ , если  $t \sqsubseteq id$ .

Данное определение позволяет интерпретировать имена  $id$  как имена ресурсов, описываемых элементами иерархии  $T_{\mathfrak{R}}^D$ , а элементы  $t \in T_{\mathfrak{R}}^D$  (не обязательно полные) — как информацию, описывающую ресурс с именем  $id$ . Отметим, что выражение  $id :: t$  отличается по смыслу от  $I(id) = t$ , поскольку во втором случае  $t$  — обязательно полный элемент.

Для работы с анонимными объектами будем пользоваться обозначением  $_ :: t$ , что может интерпретироваться как «в предметной области существует некоторый объект, соответствующий описанию  $t$ ». Обращаем внимание на то, что в случае именованных объектов одновременное наличие двух описаний  $id :: t_1$  и  $id :: t_2$  означает два описания одного и того же ресурса, в то время как одновременное присутствие анонимных описаний  $_ :: t_1$  и  $_ :: t_2$  совершенно не гарантирует того, что это описан один и тот же объект.

## § 7. Онтологии

Построенные выше конструкции позволяют дать в терминах семантического программирования формальное и абстрактное определение понятия онтологии как формальной системы, предназначенной для описания предметных областей и реализующей концептуальные схемы, приведенные выше. Начнем построение с введения языка онтологий, который строится как расширение языка  $L_t$ . Онтология строится как конкретная реализация абстрактной онтологической формы, поэтому язык конкретной онтологии появляется, как только фиксируется онтоформа. Зафиксируем такую онтоформу  $\mathbf{K}_D = \langle \mathbf{DT}, \mathbf{CN}, \mathbf{Attr}, \mathbf{ID}, \succ_D, \triangleleft_D \rangle$ . Тогда составляющими языка онтологий являются:

1. Для именованя классов используются константы  $\mathbf{CN} = \{\mathbf{cn}_1, \mathbf{cn}_2, \dots, \mathbf{cn}_p\}$ . В дальнейшем будем считать, что  $\mathbf{CN}$  содержит константу **object**, которая интерпретируется как класс всех объектов. Это означает, что для любой предметной области  $D$  любая корректная интерпретация констант из  $\mathbf{CN}$  должна сохранять свойство для любого  $\mathbf{cn} \succ_D \mathbf{object}$ . Изначальное присутствие наибольшего (а значит, наименее информативного) класса необходимо в качестве стартовой точки для построения иерархии наследования. Это, в частности, объясняет, почему в объектно-ориентированных языках программирования класс *Object* является предопределенным.
2. Чтобы иметь возможность работать с типами данных базовой модели  $\mathfrak{R} = \langle M_1, \dots, M_s; \Omega \rangle$  будем использовать в языке множество  $\mathbf{DT} = \{dt_1, \dots, dt_s\}$

констант, символизирующих основные множества базовой модели  $\mathfrak{R}$ .

3. Включим в язык онтологий бинарный предикатный символ принадлежности атрибута элементу  $\triangleleft$ , предикат определения ресурса  $::$ , а также отношение принадлежности элемента классу  $\in$ .
4. Добавим счетное множество переменных  $X = \{x_1, x_2, \dots\}$ , принимающих значения на множестве  $T_{\mathfrak{R}}^D \cup \bar{M}$ .

Получившуюся сигнатуру обозначим  $L_{Ont}^D = L_t^D \cup \mathbf{CN} \cup \mathbf{DT} \cup X \cup \{\triangleleft, \in, ::\}$ .

Сформулируем теперь понятие онтологии. Зафиксируем некоторый язык (класс формул)  $\Sigma$  сигнатуры  $L_{Ont}^D = L_t^D$ , а также некоторое его подмножество  $\Sigma_C$  (в общем определении онтологии структура языка  $\Sigma$  не уточняется). Выражение  $F(x_1, \dots, x_n)$  означает, что все свободные переменные формулы  $F \in \Sigma$  находятся среди  $x_1, \dots, x_n$ .

**Определение.** Пусть  $\mathbf{CN} = \{\mathbf{cn}_1, \mathbf{cn}_2, \dots, \mathbf{cn}_p\}$ . Назовем описанием классов набор правил

$$\begin{aligned} x &\in \mathbf{cn}_1[\mathbf{cn}_1^1, \dots, \mathbf{cn}_{k_1}^1] \square F_1(x) \\ x &\in \mathbf{cn}_2[\mathbf{cn}_1^2, \dots, \mathbf{cn}_{k_2}^2] \square F_2(x) \\ &\vdots \\ x &\in \mathbf{cn}_p[\mathbf{cn}_1^p, \dots, \mathbf{cn}_{k_p}^p] \square F_p(x) \end{aligned}$$

где  $F_i(x) \in \Sigma_C$ .

Выражение вида  $x \in \mathbf{cn}_i[\mathbf{cn}_1^i, \dots, \mathbf{cn}_{k_i}^i] \square F_i(x)$  назовем правилом, определяющим класс с именем  $\mathbf{cn}_i$  как непосредственного наследника классов  $\mathbf{cn}_1^i, \dots, \mathbf{cn}_{k_i}^i$ . Это определение должно быть согласовано с онтоформой  $\mathbf{K}_D$  в следующем смысле: если описание  $\square$  содержит правило вида  $x \in \mathbf{cn}_i[\dots \mathbf{cn}_j \dots] \square F(x)$ , то  $\mathbf{cn}_i \succ_D \mathbf{cn}_j$ , причем  $\mathbf{cn}_i$  — непосредственный наследник  $\mathbf{cn}_j$ , т. е. не существует  $\mathbf{cn}$  такого, что  $\mathbf{cn}_i \succ_D \mathbf{cn} \succ_D \mathbf{cn}_j$ .

**Определение.** Онтологией назовем тройку  $\langle \Sigma, \square, \vdash \rangle$ , где  $\Sigma$  — язык сигнатуры  $L_{Ont}^D$ ,  $\square$  — описание классов, а  $\vdash$  — финитное отношение выводимости, действующее на формулах языка  $\Sigma, \vdash \subseteq \mathbf{Fin}(\Sigma) \times \Sigma$ .

В общем случае отношение выводимости  $\vdash$  зависит от предметной области, описываемой онтологией  $\langle \Sigma, \square, \vdash \rangle$ . Его основная задача — манипуляция со знаниями, «зашифрованными» в элементах термальной надстройки. Финитность  $\vdash$  является существенным его качеством, позволяя реализовать концепцию «слабого но эффективного» вывода, действующего на множестве ресурсов, о чем говорилось в начале работы. Для  $\vdash$  существует набор аксиом, выполняющихся в любой онтологии. Этот набор описывает работу с именами ресурсов и с правилами из  $\square$ . Приведем некоторые из аксиом ( $t[t']$  означает, что  $t'$  является подтермом  $t$ ):

$$(A1) \Gamma, F(t), id :: t \vdash F(id)$$

$$(A2) \Gamma, id :: t_1, id :: t_2 \vdash id :: (t_1 \sqcup t_2)$$

$$(A3) \Gamma, n :: t[p(t')] \vdash nm :: t[p(t' \sqcup cn(\emptyset, \emptyset))],$$

$$\text{если } \mathbf{p} \triangleleft_D \langle \mathbf{cn}^*, \mathbf{cn} \rangle \in \mathbf{K}_D \text{ и } nm \in ID \cup \{', \_'\}.$$

$$(A4) \Gamma, F(t) \vdash t \in \mathbf{cn},$$

если  $x \in \mathbf{cn} \square F(x)$  — правило из  $\sqcup$

Правило *A2* является основным способом накопления информации об объекте на основании разрозненных источников. Отметим, что в этом правиле четко прослеживается роль имени как «магнита», притягивающего информацию (эта роль имени уже обсуждалась нами выше). Для анонимных объектов данное правило неприменимо. Правило *A3* позволяет учитывать при выводе информацию из онтоформы  $\mathbf{K}_D$ . Кроме общезначимых аксиом, конкретная онтология может включать аксиомы, соответствующие описываемой предметной области. Например, можно указать транзитивность и симметричность некоторого свойства из множества *Attr*:

$$\Gamma, \text{родственник}(t_1) \triangleleft t_2, \text{родственник}(t_2) \triangleleft t_3 \vdash \text{родственник}(t_1) \triangleleft t_3$$

$$\Gamma, \text{супруг}(t_1) \triangleleft t_2 \vdash \text{супруг}(t_2) \triangleleft t_1$$

и так далее.

### Заключение

В данной работе были заданы лишь некоторые базовые моменты, связанные с настройкой концепции семантического программирования на обработку информации в Интернете и не только. Многие вещи требуют значительного развития. Отметим лишь некоторые направления работ:

1. С теоретической точки зрения требуется развитие и уточнение логических структур, лежащих в основе представления информации с помощью наследственно-конечных надстроек над базовой моделью типов данных. Возможно, требуют уточнений и базовые понятия, определенные выше. С этой точки зрения наше изложение можно воспринимать как «черновик» базовых компонент теории. Уже сейчас видно, что полезными являются подходы, связанные с представлением логических конструкций в формате пространств Ю. Л. Ершова, информационных систем Даны Скотта [55], а также как системы *cc*-программирования (concurrent constraint programming [56]), в которой элементы надстройки  $T_{\mathcal{RX}}^{Attr}$  будут функционировать как отдельные процессы, коммуникация которых реализуется через общие переменные. Этот подход связан с общей проблемой построения логических исчислений в рамках ТИР, причем исчислений, интересных с практической точки зрения, позволяющих реализовывать практические задачи поиска и обработки информации в виртуальных информационных средах. Отсюда возникает вопрос о построении эффективных стратегий поиска вывода в данных логических исчислениях, а также построении соответствующих систем автоматического доказательства. Ограничим перечисление этими задачами, хотя имеется и ряд других теоретических проблем, подлежащих решению.

2. Прикладные аспекты. С нашей точки зрения, универсальность и в то же время естественность рассматриваемых конструкций позволяет надеяться на возможность использования данного диалекта семантического программирования в самых разных сферах обработки информации. В первую очередь следует упомянуть о новых подходах, связанных с обработкой документов, представленных в наиболее распространенных в сегодняшнем Интернете форматах — XML и HTML. Следует также упомянуть о

системах метаописаний ресурсов в Интернете — идее, на которой основан проект the Semantic Web. Ближе к этому стоит вопрос описания предметных областей в формате онтологий, что позволяет описывать знания в виде, удобном для использования Интернет-сервисами, в частности, теми же системами метаописаний (эта работа проводится нами в рамках системы МЕТА [7–10], которая постепенно мигрирует от решения относительно узкой задачи метаописания образовательных ресурсов на базе корпоративных стандартов IMS и Openet [37–39, 11–12] к универсальным системам описаний на основе онтологий). Отдельно стоит задача построения систем вывода новых знаний из описаний ресурсов и метаданных, а также использования этих знаний для практической работы с информационными ресурсами. По нашему представлению, здесь есть очень интересные возможности вплоть до развития принципиально новых подходов к структуре операционных систем и работы с файлами как ресурсами (эти приложения ориентированы на самый широкий круг пользователей компьютеров). Любопытно исследовать возможности ТИР для новых способов работы с ресурсами на локальных компьютерах, что можно охарактеризовать как объектный подход к построению операционных систем. И конечно, очень интересной представляется задача построения языков логического программирования нового поколения, что мы пытаемся реализовать в рамках развития функционально-логического языка Флэнг [3–4].

Можно говорить о методологических и других аспектах использования семантического программирования в глобальной информационной среде. Все это — предмет дальнейших исследований.

### Список литературы

1. *Goncharov S. S., Ershov Yu. L., Sviridenko D. I.* “Semantic programming” // Information processing, Proc. IFIP 10-th World Comput. Congress. Dublin. Vol. 10. 1986. P. 1093-1100.
2. *Goncharov S. S., Ershov Yu. L., Sviridenko D. I.* “Semantic foundations of programming” // Lecture Notes in Computer Science. Vol. 278. 1987. P. 116–122.
3. *Mantsivoda A.* Flang: A Functional-Logic Language. Lecture Notes in Computer Science, **567**, 257–270.
4. *Mantsivoda A., Petukhin V., Weimann A.* Memory Management of Constraints in Flang // Proc. of 10th Int. Conf on Logic Programming / ed. by D. S. Warren. MIT Press. 1993. P. 633–646.
5. *Манцивода А. В.* Язык Флэнг и обработка XML-документов // Труды Всероссийской научной конференции «Научный сервис в сети Интернет-2004». МГУ, 2004. С. 236–239. Электронная версия <http://www.teacode.com/public/abrau-2004-1.txt>.
6. Веб-страница Флэнг-проекта <http://teacode.com/flang>. 2002–2004.
7. Web-страница системы МЕТА: документация и дистрибутив ранней версии <http://teacode.com/meta>. 2004.
8. *Манцивода А. В., Малых А. А.* МЕТА: разработка метаописаний образовательных ресурсов // Труды Всероссийской конференции «Телематика 2003». С.-Петербург.

2003. С. 169–170.
9. *Малых А. А., Манцивода А. В.* Система МЕТА и открытые модели знаний // Труды Всероссийской научной конференции «Научный сервис в сети Интернет-2004» МГУ. 2004. С. 173–175. Электронная версия <http://www.teacode.com/public/abrau-2004-2.txt>.
  10. *Малых А. А., Манцивода А. В.* МЕТА: метаописания и образовательные пакеты // Труды Всероссийской конференции «Телематика 2004». С.-Петербург. 2004.
  11. *Открытое образование: стандартизация описания информационных ресурсов* / Горбунова Е. И., Лобачев С. Л., Малых А. А., Манцивода А. В., Поляков А. А., Солдаткин В. И.; Отв. ред. Лобачев С. Л. и Манцивода А. В. М.: РИЦ «Альфа» МГОПУ им. М. А. Шолохова, 2003.
  12. *Основы открытого образования* / Андреев А. А., Горбунова Е. И., Лобачев С. Л., Лупанов К. Ю., Малых А. А., Манцивода А. В., Солдаткин В. И.; Отв. ред. Солдаткин В. И. Т. 4. Российский государственный институт открытого образования. НИИЦ РАО. М., 2004 (в печати).
  13. *Малых А. А., Манцивода А. В., Петухин В. А.* Электронные учебные материалы: стандарты и решения // Труды Всероссийской конференции «Телематика 2002». СПб. С. 85–86.
  14. *Манцивода А. В., Петухин В. А.* Порталы, обработка структурированной информации и языки искусственного интеллекта, Труды Всероссийской конференции «Телематика 2003». СПб.. 2003. С. 168–169.
  15. *Нартов Д. С., Романова О. А., Стукушин Н. О.* Онлайн-консультация по математике // Труды Всероссийской конференции «Телематика 2003». СПб. 2003. С. 203–205.
  16. *Горбунова Е. И., Лобачев С. Л., Малых А. А., Манцивода А. В.* Подход к стандартизации в информационно-образовательной среде открытого образования, Труды Всероссийской конференции «Телематика 2003». СПб. 2003. С. 423–425.
  17. *Манцивода А. В., Малых А. А.* Метаописания и логическая структура электронных образовательных ресурсов // Труды Международного форума «Новые инфокоммуникационные технологии: достижения, проблемы, перспективы». Новосибирск. 2003. С. 73–77.
  18. *Манцивода А. В., Липовченко В. В., Малых А. А., Нартов Д. С., Петухин В. А., Романова О. А., Шивторов М. И.* Математические формулы и электронные образовательные ресурсы // Труды Международного форума «Новые инфокоммуникационные технологии: достижения, проблемы, перспективы». Новосибирск. 2003. С. 78–84.
  19. *Манцивода А. В., Петухин В. А., Шивторов М. И.* RTF, LaTeX и логическая разметка документов // Труды Всероссийской конференции «Телематика 2004». СПб. 2004.
  20. *Липовченко В. А., Манцивода А. В.* Трансляция математических формул из документов MS WORD в стандартный формат // Труды Всероссийской конференции «Телематика 2004». СПб. 2004.

21. Куроптев А. А., Манцивода А. В. Изображение математических формул в формате MathML // Труды Всероссийской конференции «Телематика 2004». СПб. 2004.
22. Малых А. А., Манцивода А. В., Романова О. А., Стукушин Н. О. QTI-плеер: онлайн-сервис поддержки тестирования. // Труды Всероссийской конференции «Телематика 2004». СПб. 2004.
23. Манцивода А. В. Сплан: «человеко-ориентированный» язык для построения формальных онтологий и спецификаций. Препринт. 200.
24. Semantic Web activity. <http://www.w3.org/2001/sw/>.
25. World Wide Web Consortium. <http://www.w3.org>.
26. Uniform Resource Identifiers (URI): Generic Syntax, 1998. <http://www.ietf.org/rfc/rfc2396.txt>.
27. Спецификация Unicode. <http://www.unicode.org/>.
28. Extensible Markup Language (XML) 1.0 (Third Edition), W3C Recommendation, 04 February 2004. <http://www.w3.org/TR/2004/REC-xml-20040204>.
29. St. Laurent, Simon XML: A Primer, second edition. Foster City, CA: M & T Books, 1999.
30. XML Schema Part 2: Datatypes, W3C Recommendation, World Wide Web Consortium, 2 May 2001. This version is <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502>.
31. Resource Description Framework (RDF). <http://www.w3.org/RDF/>.
32. Web Ontology Working Group. <http://www.w3.org/2001/sw/WebOnt/>.
33. OWL Web Ontology Language. Overview. <http://www.w3.org/TR/2004/REC-owl-features-20040210/>.
34. Horrocks I., Patel-Schneider P., van Harmelen F. From SHIQ and RDF to OWL: The making of a Web Ontology Language. <http://www.cs.man.ac.uk/~horrocks/Publications/download/2003/HoPH03a.pdf>.
35. Connolly D, van Harmelen F., Horrocks I., McGuinness D. L., Patel-Schneider P. F., Stein L. A. DAML+OIL (March 2001) reference description. W3C Note, 18 December 2001. <http://www.w3.org/TR/2001/NOTE-daml+oil-reference-20011218>.
36. The description logic handbook. Theory, Implementation and Applications. Edited by Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, Peter Patel-Schneider, 2003, Cambridge.
37. Консорциум IMS. <http://www.imsglobal.org>.
38. Спецификация IMS «Метаданные учебных объектов. Информационная модель» (IMS Learning Resource Meta-Data Information Model Version 1.2.1 Final Specification. [http://www.imsglobal.org/metadata/imsmdv1p2p1/imsmd\\_infov1p2p1.html](http://www.imsglobal.org/metadata/imsmdv1p2p1/imsmd_infov1p2p1.html).
39. Learning object metadata. IEEE Learning Technology Standards Committee. <http://ltsc.ieee.org/>.
40. Sowa J. F. Ontology, Metadata and Semiotics. <http://users.bestweb.net/~sowa/peirce/ontometa.htm>.
41. Berners-Lee T., Hendler J., Lassila O. The Semantic Web, Scientific American, May, 2001.
42. Lassila O., Swick, R. R. Resource Description Framework (RDF) Model and Syntax

- Specification. W3C recommendation, Feb. 1999.  
<http://www.w3.org/TR/1999/REC-rdf-syntax-19990222>.
43. *Brickley D., Guha R.V.* Resource Description Framework (RDF) Schema Specification. W3C proposed recommendation, Mar. 1999.  
<http://www.w3.org/TR/1999/PR-rdf-schema-19990303>.
44. RDFCore: Resource Description Framework (RDF) Schema Specification 1.0.
45. *Champin, P.-A.* RDF Tutorial, 2001.  
<http://www710.univ-lyon1.fr/~champin/rdf-tutorial/>.
46. Пространства имен в XML. <http://www.w3.org/TR/REC-xml-names>, перевод:  
<http://www.rol.ru/news/it/helpdesk/xnamsp.htm>.
47. Dublin Core Metadata Initiative (Дублинское ядро: инициативная группа по метаописаниям), <http://dublincore.org>.
48. *Hillmann D.* Using Dublin Core. Dublin Core Metadata Initiative.  
<http://dublincore.org/documents/2003/08/26/usageguide>.
49. Dublin Core projects. <http://dublincore.org/projects>.
50. *Lagoze K.* Keeping Dublin Core simple. Cross-Domain. Discovery or resource description? // D-Lib Magazin. 2001. No. 1.  
<http://www.dlib.org/dlib/january01/lagoze/01lagoze.html>
51. *Baker T.* A grammar of Dublin Core // D-Lib Magazin. 2000. No. 10.  
<http://www.dlib.org/dlib/october00/baker/10baker.html>.
52. DCMI Metadata Terms. <http://dublincore.org/documents/dcmi-terms/>.
53. Guidelines for implementing Dublin Core in XML.  
<http://dublincore.org/documents/dc-xml-guidelines/>.
54. Universal decimal classification consortium. <http://www.udcc.org>.
55. *Scott D.* Domains for denotational semantics // Lecture notes in computer science proceedings of the 9th colloquium on automata. Languages and programming table of contents. P. 577–613. 1982.
56. *Saraswat V. A.* Constraint logic programming. MIT Press, 1993.

Материал поступил в редколлегию 03.02.2005

#### Адрес автора

МАЛЫХ Антон Александрович  
Манцивода Андрей Валерьевич  
РОССИЯ, 630090, г. Новосибирск  
ул. Пирогова 2, к. 403  
тел.: (383) 339-73-29  
e-mail: [vestnikmath@nsu.ru](mailto:vestnikmath@nsu.ru)