



OSTIS-2014

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ПЕРСОНАЛЬНАЯ ОНТОЛОГИЯ ПРОФЕССИОНАЛЬНОГО ОПЫТА

Соснин П.И.*

* *Ульяновский государственный технический университет,
г. Ульяновск, Российская Федерация*

sosnin@ulstu.ru

В работе приводятся результаты исследований вопросов онтологизации персонального профессионального опыта. Специфику решений по созданию и использование персональной онтологии определяет ориентация на представление опыта в виде системы моделей прецедентов, освоенных и созданных субъектом деятельности в реализации совокупности проектов. Модели прецедентов построены по образцу интеллектуально обработанных «деятельностных рефлексов» с использованием вопросно-ответных рассуждений.

Введение

Деятельность – это форма жизни, в основу которой природа положила интеллектуально обработанные условные рефлексы, формирующие то, что позволяет ввести в типовые взаимодействия человека с окружающей средой «вторую сигнальную систему» [Данилова и др. 1997], то есть естественный язык (ЕЯ). Приписывание знаков ЕЯ обстоятельствам, которые распознаются с помощью органов чувств в определённой окружающей обстановке, открывает возможность для абстрагирования от этой обстановки до знакового образа, позволяющего управлять соответствующим типовым поведением человека.

Такое типовое поведение принято называть «прецедентом», а знаковый образец, по которому оно осуществляется «моделью прецедента». То, что называют «опытом человека» включает громадное количество «прецедентов», взаимодействие с которыми (для развития, поиска, доступа, настройки, рационализации и других целей) невозможно без их систематизации.

Разумно различать неявную *естественную систематизацию*, за существование которой отвечают естественные механизмы опыта человека, *естественно-искусственную систематизацию*, ответственность за которую несёт освоенная человеком система понятий, и *искусственную систематизацию*, материализация и использование которой осуществляется с помощью искусственных сред, например, компьютеризованного типа.

Первые два вида систематизации имеют отношение только к человеку, а третий вид

допускает его применение не только человеком, а, например, и программными (интеллектуальными) агентами, в различных приложениях. Именно к третьему виду относятся искусственные средства систематизации, получившие название «онтологий». За созданием любой «онтологии» всегда стоят задачи, для решения которых используются взаимодействия с опытом и его моделями.

В статье предлагается подход к формированию персональных онтологий, обслуживающих взаимодействие конкретного индивида с доступным ему опытом. Подход материализован инструментально в версии OwnWIQA вопросно-ответной моделирующей среды WIQA (Working In Questions and Answers) [Соснин и др., 2012].

1. Предварительные основы

Предлагаемый подход развивает наши исследования в проблемной области «Формирование и использование прикладных онтологий», результаты которых раскрыты в публикациях [Sosnin, 2010], [Соснин и др., 2010].

Специфику этих исследований, ориентированных на прикладные онтологии в проектировании автоматизированных систем (АС), определяют следующие установки:

1. Онтология конкретной прикладной области должна строиться в процессах проектирования АС, относящихся к этой области.

2. Функции основного информационного источника в построениях прикладных онтологий должны выполнять рассуждения лиц, вовлечённых в

процессы проектирования и использования их результатов.

3. То, что включается в онтологию, должно прямо или опосредовано найти материальное воплощение в процессах проектирования и/или их продуктах.

4. Включение онтологии в состав средств, используемых в разработках АС и их применениях, должно конструктивно приносить ожидаемые позитивные эффекты.

Отметим, что среди различных типов рассуждений был выделен класс вопросно-ответных рассуждений (QA-рассуждений), осуществляемых в процессах решения проектных задач. Выбор этого класса был обусловлен диалоговой природой сознания, обеспечивающего доступ к наличному опыту.

Именно отмеченная специфика установок привела к созданию инструментально-технологического комплекса WIQA.Net, на базе которого разработан ряд его приложений, например, для концептуального проектирования АС, документирования и обучающего сопровождения. В концептуальном проектировании «онтология проекта» применялась в оперативном предикатно-онтологическом контроле проектных решений [Соснин и др., 2012].

Еще одним приложением, разработанном на основе WIQA.Net и использующем этот комплекс, является «База Опыта проектной организации», в состав которой включена и её онтология.

Так что реальные построения и применения прикладных онтологий привели к решению исследовать возможности построения и использования персональных онтологий. Представленные выше установки были сохранены, но в их приложении не только к задачам

проектирования, которые по силам одному человеку, но и к освоению опыта решения профессиональных задач, с которыми сталкивается конкретный индивид. Этот класс задач в разработке персонального инструментария OwnWIQA было решено считать подобными задачам проектирования, которые можно решать в операционной среде OwnWIQA.

К установкам, модифицированным для построения OwnWIQA, была добавлена ещё одна:

5. Владелец персональной онтологии является и разработчиком и пользователем продуктов, при создании которых он применяет построенную (и освоенную) им онтологию.

На разработку инструментария OwnWIQA, а также методологического обеспечения его применений существенное воздействие оказали публикации в области эмпирической программной инженерии, в первую очередь, раскрывающие моделирование и использование профессионального опыта [Basili et al. 2001], вопросно-ответных механизмов доступа к базам опыта [Henninger 2003] и научного подхода к экспериментированию в человеко-компьютерных средах [Sjoberg et al. 2007].

2. Вопросно-ответная память

В соответствии с установками предлагаемого подхода персональная «онтология» предназначена для систематизации моделей прецедентов, которые освоены индивидом и используются им в решении задач $Z=\{Z_k\}$, относящихся к области его профессиональных интересов.

В спецификациях «онтологии» и её материализации принципиальное место занимает вопросно-ответная память (QA-память), обобщённо представленная на рисунке 1.

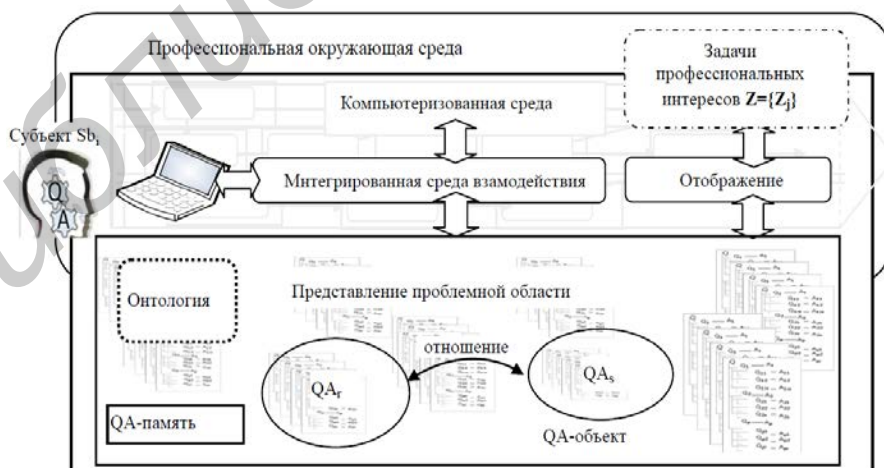


Рисунок 1 – Вопросно-ответная память инструментария OwnWIQA

QA-память – это подсистема инструментария OwnWIQA, предназначенная, в первую очередь, для моделирования QA-рассуждений, используемых индивидом (субъектом Sb_i) в процессах решения

задач. Конкретная QA-модель состоит из связанной совокупности моделей вопросов $\{Q_i\}$ и соответствующих им моделей ответов $\{A_i\}$, причём,

каждая из таких моделей загружена в определённую ячейку QA-памяти.

Каждая из ячеек QA-памяти создаётся оперативно для регистрации и хранения атрибутики очередной модели типа Q или A. Атрибутика включает базовый набор атрибутов $B(a_1, a_2, \dots, a_N)$, к которому владелец OwnWIQA (или короче I) может добавить полезные дополнительные атрибуты $AA(aa_1, aa_2, \dots, aa_M)$. Совокупность базовых атрибутов с операциями над ними образует интерактивный объект типа Q или A. Ответственность за операции над объектом, в которых используется дополнительная атрибутика, возлагается на субъекта Sb_i .

В число базовых атрибутов ячейки QA-памяти включены: уникальное имя модели (тип и индекс, приписываемый автоматически), например Q1.1.2, способное выполнять роль адреса ячейки; идентификатор создателя модели, то есть имя I; знаковая модель Q или A, в форме символьной строки; момент времени создания или модификации; имя проекта и другие атрибуты. Обобщённое представление Q-объекта приведено на рисунке 2.



Рисунок 2 – Интерактивный Q-объект

Объекты A-типа имеют подобную структуру. Приписывание дополнительных атрибутов и программирование дополнительных операций осуществляется субъектом Sb_i в процессах решения задач, если это полезно (что отражено пунктирными границами). Кроме того, в инструментарий OwnWIQA встроены средства для решения ряда системных задач, в решениях которых применены дополнительная атрибутика с дополнительными операциями. К числу таких задач относятся, например, задачи формирования документов по шаблонам на базе QA-рассуждений, зарегистрированных в QA-памяти.

Отметим, что к любому интерактивному объекту можно прикрепить файлы, например, графические файлы, которые будут включаться в документы.

И всё же особой целостностью обладает пара ячеек QA-памяти, в которых содержатся взаимно дополнительные Q-объект и A-объект. С такой парой в QA-памяти связывается QA-объект, позволяющий представлять «предикативность» и «актуальное членение» в знаковых описаниях, например, «обстоятельств окружающей среды». За такой целостностью «взаимодополнительность вопроса и ответа», что позволяет приписывать характеристику «да» (например, истинности или

правдоподобия) согласованным текстовым представлениям вопроса Q_i и ответа A_i .

С другой стороны, любая коммуникативная единица, представленная знаками языка, например «предложение», открыта для её актуального членения на «тему» и «рему» [Иванов. 2010], указывающую на то «информационно новое», для выражения которого предложение создано. Актуальное членение проще всего выразить и зарегистрировать с помощью согласованных вопроса Q_i и ответа A_i .

Следовательно, с помощью QA-объекта можно представить в QA-памяти «переменную» (Q-объект) и её «значение» (A-объект), как для любого простого типа, так и для любого составного типа. Более того, «переменной» определённого типа с помощью дополнительных атрибутов можно приписать «характеристики типа», общепринятые в языках программирования, а также полезные «семантические характеристики».

За приписыванием «переменной» её «значения» стоит «предложение», с которым (с помощью дополнительных атрибутов) можно связать определённую «модальность» или их совокупность, например, приписав «значению» его «неопределённость», например, с помощью вероятностной меры.

А значит, в QA-памяти информационно представимы «объекты» и «обстоятельства» окружающей среды, причём с привязкой ко времени и с учётом измерений и идентификации их характеристик. Отображение двух объектов среды и связи между ними в QA-памяти обобщённо представлены на рисунке 1.

В QA-памяти представимы не только объекты предметной области $Z=\{Z_k\}$, но и процессы, структурированные алгоритмически. Так, например, шаг алгоритма, выраженный в виде псевдокодированного оператора, может быть загружен в поле описания Q-объекта. Ответом на такую интерпретацию вопроса логично считать результат выполнения оператора.

Отметим, что в общем случае «вопрос» может включать «подчиненные вопросы», что приводит к иерархически связанным совокупностям QA-объектов, представленных на рисунке 3.

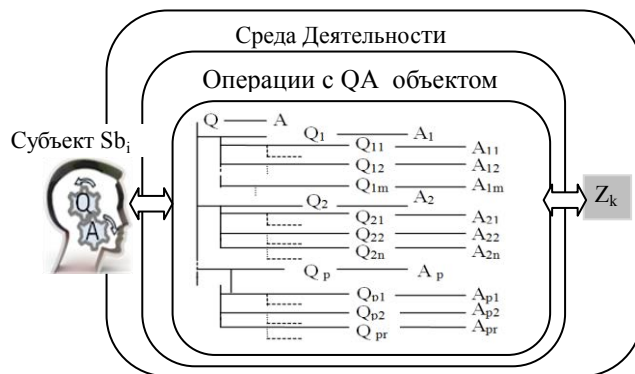


Рисунок 3 – Составной QA-объект

Отметим, что структура и содержание QA-объектов, определяются теми (профессиональными) задачами, которые приходится решать субъекту Sb_i . В частности, в QA-памяти представимы «требования» (requirements) и их «спецификация» в «проектах» или «материализация» в «продуктах», в создание которых субъект Sb_i внёс или вносит свой вклад, применяя доступный ему опыт.

3. Онтологизация персонального опыта

3.1. Ориентация на прецеденты

QA-память инструментальной среды OwnWIQA специфицирована и материализована так, чтобы в её ячейках можно было представлять решения задач $\{Z_k\}$, за которыми стоят «прецеденты», в осуществлении которых участвует субъект Sb_i . Более того, для «моделей прецедентов», которые хранятся в QA-памяти, должны использоваться нормативные формы, обобщённая структура которых приведена на рисунке 4.



Рисунок 4 – Модель прецедента

Модель прецедента P_k включает: текстовую составляющую P_k^T , в виде постановки задачи; логическую составляющую P_k^L , формула которой представлена на рисунке 4; вопросно-ответную модель P_k^{QA} задачи Z_k ; графическое представление прецедента P_k^G ; исходный псевдокод P_k^I и исполняемый код P_k^E .

Нормативная модель прецедента построена таким образом, чтобы она раскрывала концептуальное содержание задачи и представляла её концептуальное решение (псевдокод решения). По этой причине в модели принципиальное место занимают конструкции на естественно-профессиональном языке L^P предметной области $\{Z_k\}$. Основными из этих конструкций являются текст T_k постановки задачи P_k^T и предложения $\{P_{ik}\}$ в модели P_k^{QA} , регистрирующей вопросно-ответный анализ этой задачи, в результате которого на все важные вопросы по её решению получены ответы, достаточные для построения логической схемы P_k^L и алгоритмического решения P_k^I .

В комплексе OwnWIQA сохранена ориентация активности его пользователя на «проекты», что позволяет выделять на множестве задач $\{Z_k\}$ их взаимосвязанные совокупности $S_m\{Z_k^m\}$, за каждой из которых стоит определённый «проект»,

выполненный или выполняемый субъектом Sb_i . Использование структуризации активности в единицах типа «проект» позволила заимствовать из комплекса WIQA.Net ряд средств для организации и управления процессами (псевдо) параллельного решения задач, а также заимствовать опыт создания и использования «онтологий проектов» [Соснин и др., 2012]. Так что в «персональной онтологии» $O_k=I(\{O_k^m\})$ интегрируются совокупность «онтологий проектов» $\{O_k^m\}$ субъекта Sb_i .

3.2. Лексика персональной онтологии

С множеством задач $\{Z_k\}$, решения которых ориентированы на прецеденты, связана определённая совокупность $\{T_k\} \cup \{P_{ik}\}$ употреблений языка L^P . Именно этот лингвистический материал в построениях персональной онтологии профессионального опыта должен занимать принципиальное место.

Следует отметить, что тексты и предложения из множеств $\{T_k\} \cup \{P_{ik}\}$ создаются в результате согласованного уточнения их версий, в которых применялась не только лексика моделей прецедентов $\{P_k\}$, но и лексика естественного языка L в том объёме, которым владеет субъект Sb_i . А значит из языка L логично отбирать всё то, что будет полезным для решения задачи онтологической систематизации множества моделей $\{P_k\}$.

В комплексе OwnWIQA для такого отбора используется толковый «Словарь», который строится субъектом Sb_i в процессе создания системы моделей $S(\{P_k\}, O_k)$, включающей его персональную онтологию O_k . В этом словаре содержатся и понятия онтологии, требующих их определения. Так что лексика онтологии формируется из лексики моделей прецедентов и лексики, используемой в толковом «Словаре».

3.3. Формирование приращений онтологии

В структуре персональной онтологии O^k выделяются взаимосвязанные неформальная и формализованная составляющие. Неформальная часть состоит из понятий $\{N_q\}$ онтологии с их толкованиями (на базе языков L и L^P), представленными в статьях «Словаря».

В каждой статье второй части «имя статьи» N_q связывается со следующими списками:

- списком $(g_{q1}, g_{q2}, \dots, g_{qn})$ имён дифференцирующих признаков, согласованным с определением имени N ;
- набором списков $\{(N_s, N_v, \dots, N_w)\}$ имён других статей, каждый из которых связан с определённым типом систематизации моделей прецедентов.

Выбранная версия представления онтологии $O^k=I(\{O_k^m\})$ упрощает её оперативное создание в процессе решения задач $\{S_m\{Z_k^m\}\}$, так как типовым приращением для O^k является O_k^m . Кроме того, каждая онтология O_k^m развивается за счёт приращений, основными из которых являются

имена объектов, зарегистрированных субъектом Sb_i в ячейках QA-памяти. По этой причине, формализованное содержимое онтологии O_k^m адресуется, что упрощает его включение в программную обработку, например, для доступа к моделям прецедентов.

В инструментальной среде WIQA.Net для выявления приращений использовался предикатно-онтологический контроль текстовых описаний проектных решений в определённых «проектах». Предложения такого описания (автоматизированно) переводились на прологоподобный язык, после чего выделенные предикаты проходили проверку на соответствие текущему состоянию онтологии проекта. Отрицательный результат проверки указывал либо на ошибку, либо на необходимость ввести в онтологию приращение. Средства и механизмы такого контроля детально раскрыты в публикациях [Соснин и др., 2010] и [Соснин и др., 2012].

В разработке комплекса OwnWIQA было решено отказаться от выявления приращений онтологии за счёт предикатно-онтологического контроля в связи с его сложностью (использование визуализируемого псевдофизического силового взаимодействия между членами предложений в текстах описаний) для потенциальных пользователей OwnWIQA.

Однако визуализацию предикатной структуры предложений было решено сохранить, предоставив субъекту Sb_i возможности её графического (block-and-line) представления (с помощью специализированного графического редактора). Для

выявления предикатной структуры субъект Sb_i может использовать морфологический анализатор, а в проверках механизмы онтологической фильтрации.

4. Средства создания персональной онтологии

Основным предназначением инструментария OwnWIQA является обслуживание концептуальной активности субъекта Sb_i в его профессиональной деятельности. Ориентация на профессиональную деятельность означает только то, что решаемые субъектом Sb_i задачи группируются и связываются в образования, которым можно приписать статус «проекта». Такая ориентация вводит *связность* в совокупности моделей прецедентов, а значит и *системность* в доступный для Sb_i опыт, так как часть этого опыта представлена моделями прецедентов.

Ориентация в OwnWIQA на разработку проекта привела к решению связать типовую единицу системности профессионального опыта с онтологией проекта, а интеграцию таких онтологий, то есть интеграцию $\{O_k^m\}$, связать с персональной онтологией O_k профессионального опыта субъекта Sb_i . Это же решение позволило отделить построение онтологий от действий по их интеграции, что и привело к нормативной концептуальной структуре «онтологии проекта», представленной на рисунке 5 в виде, который принято называть framework.

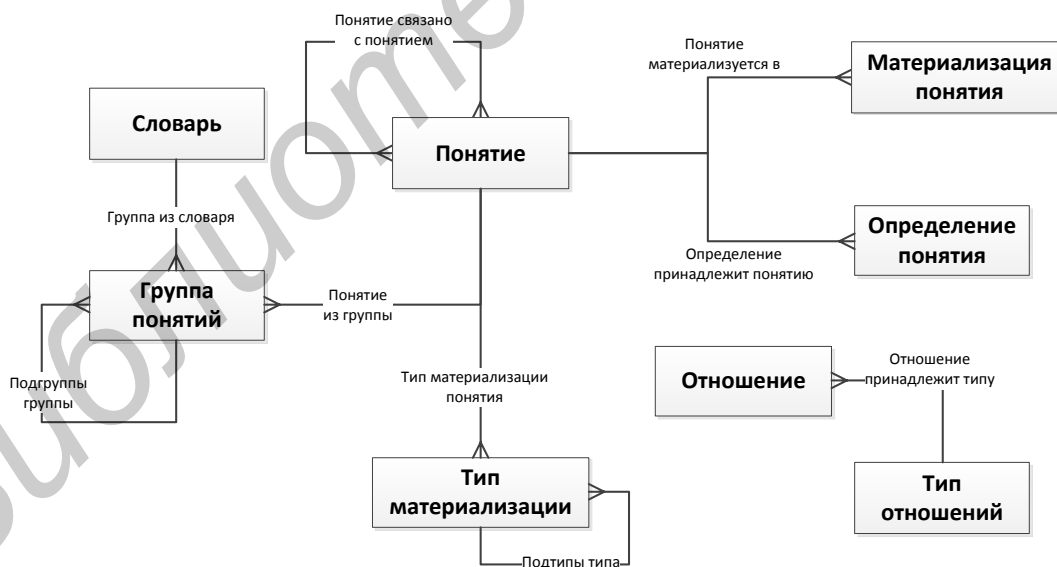


Рисунок 5 – Концептуальная структура онтологии

Так как о назначении и содержании «Словаря» было сказано выше, то пояснения в структуре необходимы только для «отношений» и «материализаций».

С различными «отношениями» в структуре типовой «онтологии проекта» связаны различные типы систематизации, которые можно ввести в

конкретную онтологию O_k^m . Приписывание отношениям имён и типов осуществляется субъектом Sb_i в операционной обстановке, фрагмент которой приведён на рисунке 6. Полезность типов, представленных в интерфейсной форме, была выявлена при создании и использовании «онтологий проектов» в инструментальной среде WIQA.Net. Отметим, что отобранный набор типов

существенно богаче, чем те наборы типов, которые традиционно используются при создании онтологий.

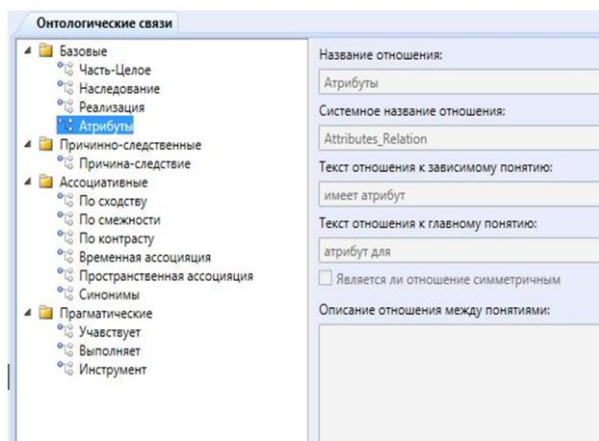


Рисунок 6 – Приписывание онтологических связей

Отметим, что отношениям можно не только приписать имена, выводящие на их семантику, но и прокомментировать. Такая возможность особо полезна для прагматических отношений, например, инструментального типа (связывающего объект с инструментами для его обработки или процесс со средствами его реализации).

«Материализация» введена в структуру онтологии для референциального связывания того, что стоит за понятием с «референтом», на которое понятие указывает при его употреблении в текстах или рассуждениях. Часть списка типовых «референтов» человеко-компьютерной среды, а также операционная среда их выбора и спецификации приведены на рисунке 7.

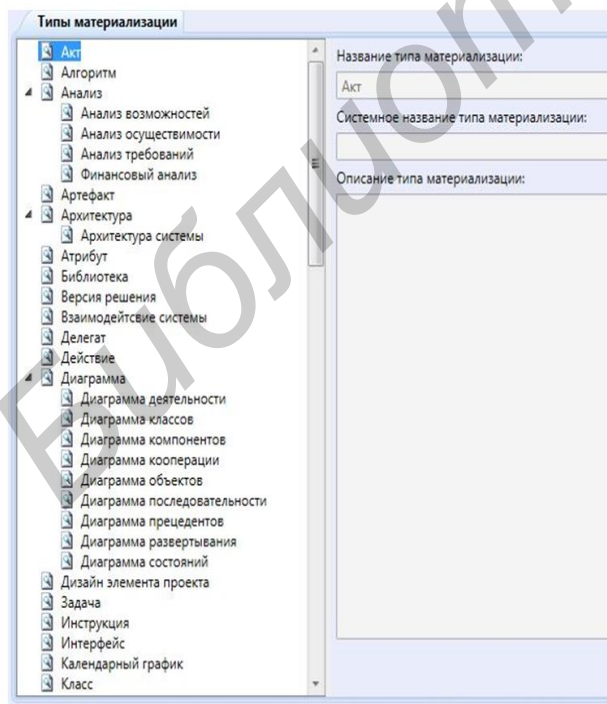


Рисунок 7 – Приписывание материализации

Учёт «материализации» повышает

ответственность субъекта Sb_i за выбор слов, используемых им в решении задач, способствует предотвращению ошибок, а также вводит в онтологию дополнительную систематизацию. Отметим, что список «типов материализации» открыт для включения новых составляющих.

Отметим и то, что для выполнения конкретной референциальной функции необходимо представить «референт» в среде OwnWIQA определённым файлом или указать «местоположение» референта в таком файле. Спецификация референции регистрируется с помощью интерфейсной формы, представленной на рисунке 8.

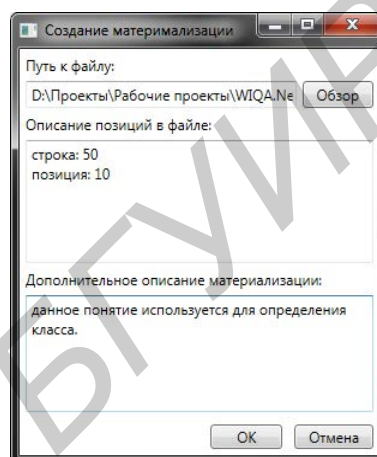


Рисунок 8 – Спецификация референта

Для того чтобы перейти к файлу, в котором материализовано понятие необходимо:

1. Подключить онтологию проекта, в котором находится понятие
2. В дереве онтологий выбрать необходимое понятие
3. Перейти на вкладку «Материализация»
4. В списке материализаций выбрать материализацию, к файлу которой необходимо перейти
5. Вызвать контекстное меню
6. Выбрать пункт «Перейти к файлу»

Такого рода инструкции имеются для типовых процедур OwnWIQA. В то же время для наиболее часто используемых процедур, возможно их выполнение за счёт активации определённых «горячих клавиш». Так, например, для перехода к файлу достаточно активировать клавиши $Ctrl + G$.

5. Реализация структур данных

Возможности создания и использования персональной онтологии в среде OwnWIQA обеспечиваются структурой данных, концептуальная схема которой приведена на рисунке 9 в форме, которая реализована в двух версиях, одна из которых ориентирована на создание онтологии и её использование субъектом Sb_i , а вторая открыта для её использования в псевдокодовых программах.

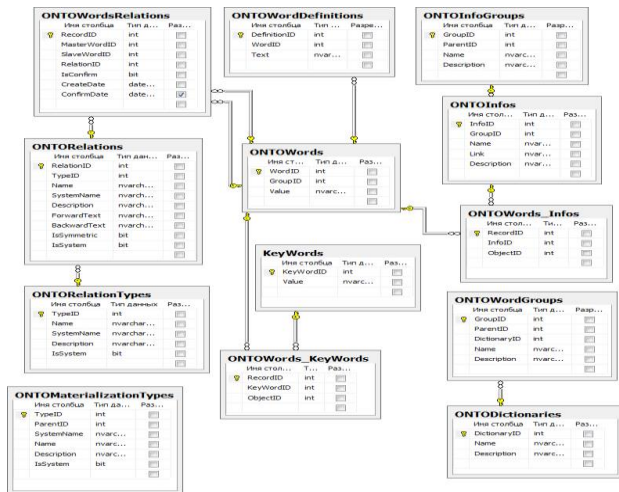


Рисунок 9 – Структура данных онтологии

Схема приводится с демонстрационными целями (без пояснений и спецификаций) только для того, чтобы показать масштабность информационной поддержки взаимодействий с онтологией. Её реализация по первой версии проведена по образцу программирования приложений с базами данных. Все интерфейсы, показанные выше, относятся к реализации именно этой версии.

Вторая версия представляет собой «клон» первой версии, загруженный как системный проект в QA-память комплекса OwnWIQA. Такая возможность обусловлена тем, что основу онтологии определяет иерархическая структура системы понятий. Системные отношения между понятиями представляются списками, которые регистрируются в полях описания соответствующих ячеек QA-памяти.

Подобный приём с клонированием фрагментов Базы данных используется в комплексе WIQA.Net для данных об организационной структуре коллектива проектировщиков [Sosnin, 2013].

6. Использование персональной онтологии

В комплекс OwnWIQA, встроены два базовых варианта применения персональной онтологии, первый из которых осуществляется в традиционных формах человеко-компьютерного взаимодействия. В рамках этого варианта субъект Sb_i имеет визуальный доступ к систематизации, которую он сам построил для совокупности моделей прецедентов, использованных или используемых им в определённых проектах. В таком взаимодействии принципиальным является проверенное и контролируемое использование лексики, согласованное с пониманием прецедентов.

Второй вариант открывает дополнительные возможности, во-первых, вопросно-ответного моделирования взаимодействий с онтологией, например, для проверки корректности её содержания и связей, и, во-вторых, для псевдо-

кодового программирования задач, в которых полезно применение онтологии.

К числу таких задач относится, например, поиск подходящей модели прецедента в сложившихся условиях. В OwnWIQA решение такой задачи возложено на программного агента, запрограммированного на псевдо-кодовом языке WIQA [Sosnin, 2013]. Обобщённая структура этого агента приведена на рисунке 10.

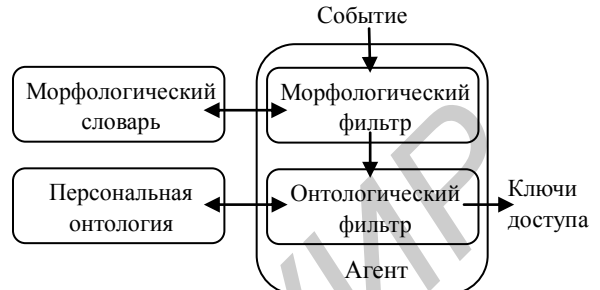


Рисунок 10 – Структура агента

Агент считывает с «доски поиска» описание события, затребовавшего доступ к системе моделей прецедентов. Текст этого описания обрабатывается «морфологическим фильтром», который выделяет из описания список потенциальных «ключей доступа» к Базе прецедентов. «Онтологический фильтр» исключает из этого списка те лексемы, которые отсутствуют в онтологии.

7. Информационные источники

На формирование персонального профессионального опыта принципиальное воздействие оказывают информационные источники различных типов, доступные в компьютеризованных средах. Так, например, не следует изобретать собственные определения понятий, для хорошо освоенных проблемных областей. Их следует заимствовать из тех источников, которым можно доверять.

Именно такого рода понятия следует накапливать и систематизировать в той части онтологии, которая является общей для проектов субъекта Sb_i . Следует заметить, что, заимствуя устоявшиеся понятия, следует сохранить их связь с соответствующим источником или источниками. Такая функция в OwnWIQA возложена на компоненту «Информационные источники», концептуальная структура которой приведена на рисунке 11.



Рисунок 11 – Информационные источники

В концептуальной структуре присутствует блок «сущность», экземпляр которой может быть не только понятием, но и описанием определённого объекта. Независимо от того, что представляет собой экземпляр сущности, ему можно приписать «Ключевое слово» или совокупность таких слов, например для доступа в Интернет.

Заключение

В статье представлены средства, предназначенные для создания и использования персональной онтологии профессионального опыта в инструментально-моделирующей среде OwnWIQA. Инструментарий ориентирован на пользователя, который (в своих собственных проектах) решает задачи, а также представляет их решения в виде моделей прецедентов для повторного использования в будущих проектах. Он же, по ходу работ, систематизирует модели прецедентов, строя персональную онтологию опыта, соответствующую освоенным прецедентам.

Спецификации персональной онтологии специально согласованы с деятельностью в рамках проектов, что позволяет: отдельно создавать онтологии проектов и объединяя их в единую систему; расширить набор типов онтологической систематизации; вводить референциальные связи составляющих онтологии с составляющими моделями опыта; использовать содержимое онтологии в программировании задач.

Библиографический список

- [Данилова и др. 1997] Данилова Н.Н., Крылова А.Л. Физиология высшей нервной деятельности / Н.Н. Данилова А.Л. Крылова – М.: Учебная Литература, 1997.
- [Иванов. 2010] Иванов Н. В. Актуальное членение предложения в текстовом дискурсе и языке/ Н. В.Иванов // Ozon.ru, Азбуковник, 2010
- [Соснин и др., 2010] Соснин П.И., Шамшев А.Б. Комплекс средств контроля семантики проектных задач и проектных решений / П.И. Соснин, А.Б. Шамшев // «Автоматизация процессов управления», 2010, № 3, – С. 55-62.
- [Соснин и др., 2012] Соснин П.И., Маклаев В.А. Инструментальные средства для спецификации концептуализаций в проектировании автоматизированных систем/ П.И. Соснин, В.А. Маклаев // Онтология проектирования, 2012. № 2, – С. 39-52.
- [Basili et al. 2001] Basili V. R., Lindvall M. Costa P., Implementing the experience factory concepts as a set of experience bases/ V. R. Basili, M. Lindvall, P. Costa// In Proc. of the International Conference on Software Engineering & Knowledge Engineering, 2001, pp. 102-10
- [Henninger et al. 2001] Henninger S., Tool Support for Experience-based Software Development Methodologies/ S. Henninger// Advances in Computers, vol. 59, 2003, pp. 29-82
- [Sjoberg et al. 2001] Sjoberg D.I. K., Dyba T., Jorgensen M., The Future of Empirical Methods in Software Engineering Research/ D.I. K. Sjoberg, T. Dyba, M. Jorgensen // In Proc. of Workshop Future of Software Engineering, 2007, pp. 358-378.
- [Sosnin, 2010] Sosnin P., Creation and Usage of Project Ontology in Development of Software Intensive Systems/ P. Sosnin// Polibits. – 2010 – № 2, pp. 132-146.
- [Sosnin, 2013] Sosnin P., A Scientifically Experimental Approach to the Simulation of Designer Activity in The Conceptual Designing of Software Intensive Systems/ P. Sosnin// IEEE ACCESS, vol. 1, 2013. C.488-504.

A PERSONAL ONTOLOGY OF PROFESSIONAL EXPERIENCE

P.I. Sosnin*

Ulyanovsk State Technical University, Ulyanovsk, Russia

sosnin@ulstu.ru

The paper presents an ontological systematization of the experience which supports a professional activity of a person. Specificity of the offered approach (to the creation of the personal ontology) is defined by using the precedent' model as a basic type of experience units. Models of precedents are constructed in accordance with the normative scheme during the project activity of the person. Solving the project tasks the person uses question-answer reasoning that supports access to the necessary experience.

Introduction

Human experience has a naturally artificial nature that is evolved in interactions of the person with surrounding. Such interactions are impossible without using the natural language providing a creation of precedents as typical units of the person behavior. Moreover, the language helps to form precedents' models and to register them in the experience. The paper is aimed at a systematization of the professional experience with the help of the personal ontology.

Main Part

Offered approach is based on a reflection of the activity area to its presentation in a question-answer memory. In this case, all tasks that are solved in interactions with experience can be presented in a uniform way. The memory of such a type correlates with the hierarchical structure of the ontology.

The personal ontology is created and used by the person in accordance with a precedent framework which is adjusted on any task being solved. Moreover, any component of the project ontology includes the referential relation with the corresponding component of the precedent model.

The project ontology is also created on the base of the normative framework. This framework is developed as a number of operational windows which support the work of the person with the dictionary, relations of systematizations and means of references between ontology and models of precedents.

Conclusion

Specifications of the personal ontology are coordinated with activity within the projects that allow: separately to create project ontology for embedding it to the integral system; to expand a set of types of ontological systematizations; to use the referential relations among ontology and experience models; to use components of the ontology in programming of tasks.