

РАЗРАБОТКА МНОГОАГЕНТНОЙ ТЕХНОЛОГИИ УПРАВЛЕНИЯ КОНТЕКСТОМ В ОТКРЫТОЙ ИНФОРМАЦИОННОЙ СРЕДЕ[♦]

А. В. СМЕРНОВ¹, Т. В. ЛЕВАШОВА, Н. Г. ШИЛОВ, М. П. ПАШКИН,
А. А. КРИЖАНОВСКИЙ, А. М. КАШЕВНИК, А. С. КОМАРОВА

Санкт-Петербургский институт информатики и автоматизации РАН

СПИИРАН, 14-я линия ВО, д. 39, Санкт-Петербург, 199178

¹<smir@iias.spb.ru>

УДК 658.519.011.56

Смирнов А. В., Левашова Т. В., Шилов Н. Г., Пашкин М. П., Крижановский А. А., Кашевник А. М., Комарова А. С. **Разработка многоагентной технологии управления контекстом в открытой информационной среде** // Труды СПИИРАН. Вып. 3, т. 1. — СПб.: Наука, 2006.

Аннотация. Разработаны, методология и исследовательский прототип многоагентной технологии управления контекстом в открытой информационной среде в рамках интеллектуальной поддержки принятия решений. Показано использование формализма объектно-ориентированных сетей ограничений для представления знаний. Определен набор технологических и проблемно-ориентированных агентов для решения задач управления контекстом в открытой информационной среде, разработаны модели и сценарии их взаимодействия. Исследовательский прототип протестирован на примере комплексной задачи конфигурирования мобильного госпиталя в ситуации техногенной катастрофы. — Библ. 47 назв.

UDC 658.519.011.56

Smirnov A. V., Levashova T. V., Shilov N. G., Pashkin M. P., Krizhanovsky A. A., Kashevnik A. M., Komarova A. S. **Development of Multiagent Technology for Context Management in Open Information Environment** // SPIIRAS Proceedings. Issue 3, vol. 1. — SPb.: Nauka, 2006.

Abstract. Methodology and research prototype of multiagent technology for context management in an open information environment for intelligent decision support have been developed. Formalism of object-oriented constraint networks used for knowledge representation is presented. A set of technological and problem-oriented agents used for accomplishing purposes of context management in open information environment is defined. Models and scenarios of agent interactions are developed. The prototype is tested using a case study for a complex task of portable hospital configuration in an emergency situation of a disaster event. — Bibl. 47 items.

1. Введение

Интенсивное развитие информационных и сетевых технологий сделало возможным обмен различными информационными составляющими (данными, информацией и знаниями) и их использование различными объектами, применяющими данные технологии и функционирующими в открытой информационной среде. Процессы обмена привели к накоплению информации и росту количества и объема информационных хранилищ. В результате получение актуальной (релевантной, доступной и достоверной) информации стало затруднительным и долговременным процессом.

Использование контекста дает возможность выявлять, какая информация является актуальной в той или иной ситуации. Актуальность информации может определяться по отношению ко всем объектам, участвующим в процессах информационного обмена: по отношению к пользователю, к прикладной про-

[♦]Работа выполнена при финансовой поддержке РФФИ (проекты № 05-01-00151 и 06-07-89242), Президиума РАН (проект № 2.35), ОИТВС РАН (проект № 1.9) и СПбНЦ РАН (проект № 78 региональной научной программы 2006 года).

грамме или системе, к решаемой задаче и т.п. В зависимости от того, по отношению к чему, определяется актуальность информации, выделяют различные виды контекста. Например, могут быть определены физический контекст, обусловливаемый местоположением и временем; контекст окружающей среды; индивидуальный контекст, характеризующий некоторую личность; социальный контекст и т.п. [1–6].

Одно из свойств контекста состоит в том, что он ограничивает [1, 2, 7] возможные варианты поведения в данной ситуации, предлагая наиболее эффективные варианты поведения. Под поведением понимается любая деятельность, связанная с процессами взаимодействия в конкретной ситуации.

Управление контекстом позволяет интегрировать несколько контекстов в один, обеспечивая тем самым получение актуальной информации от всех участников взаимодействия при решении общей задачи в рамках некоторой ситуации. И, соответственно, позволяет распространять актуальную информацию заинтересованным в ней участникам.

На основании анализа компонентов открытой информационной среды, ее архитектуры и функций интеллектуальных систем поддержки принятия решений предложена методология построения многоагентной технологии управления контекстом в данной среде, апробированная в виде исследовательского прототипа. Методология заключается в динамической интеграции контекстов объектов взаимодействия системы поддержки принятия решений (пользователя, запроса пользователя, приложения и окружающей среды) в контекст задачи с последующим ее решением как задачи удовлетворения ограничений. Данная методология была реализована в виде исследовательского прототипа многоагентной технологии управления контекстом в рамках среды «Интеграция» и протестирована на примере формирования мобильного госпиталя в ситуации (*контексте*) техногенной катастрофы.

2. Разработка теоретических основ управления контекстом в открытой информационной среде

Основным назначением контекста является предоставление релевантной и достоверной информации объектам, участвующим во взаимодействии, для решения конкретной задачи (достижения конкретной цели). Таким образом, интерпретируя используемый объектами контекст, можно получить релевантную информацию для решения данной задачи [7, 8]. В данной работе под контекстом понимается модель, описывающая знания, релевантные задаче пользователя и совместно используемые компонентами открытой информационной среды в ходе решения данной задачи.

В рамках разработанной технологии открытая информационная среда рассматривается состоящей из следующих компонентов: (i) множества конечных пользователей, (ii) информационных ресурсов, включающих в себя источники данных, информации и знаний и (iii) набора средств и методов обработки содержимого информационных ресурсов.

В результате обобщения результатов исследований типов и свойств контекстов [2–7, 9–14], способов представления контекста [1, 10, 12, 15–17] и задач, решаемых при управлении контекстом [2, 3, 7, 10, 13, 18], были специфицированы требования к управлению контекстом для интеллектуальной поддержки принятия решений в открытой информационной среде.

2.1. Спецификация требований к управлению контекстом для интеллектуальной поддержки принятия решений

Управление контекстом занимается вопросами организации контекстной информации, адаптации контекстов к данным условиям и задачам и выявления отношений между контекстами [2].

Организация контекстной информации включает в себя сбор, интерпретацию, хранение и распространение контекстной информации в динамическом режиме [3]. Для адаптации контекста к конкретным условиям известны два варианта. Первый вариант опирается на заранее подготовленную абстрактную модель контекста, которая может быть конкретизирована для заданных условий [13]. Во втором варианте предлагается создавать абстрактный контекст из текущего используемого контекста с дальнейшей конкретизацией абстрактного контекста для других условий [10]. Отношения между контекстами включают в себя отношения типа: отношений включения, когда один контекст полностью или частично содержит в себе другой; иерархические отношений, устанавливаемых между контекстами по мере уточнения (конкретизации) одним контекстом другого [7]; отношения выводимости, возникающие, когда определяемая контекстная информация логически выводится из известной контекстной информации [18].

На основании перечисленных задач, возникающих при управлении контекстом, выдвигаются следующие требования к управлению контекстом.

С точки зрения способа *представления контекста*, контекст должен быть описан стандартизированными способами, обеспечивающими независимость способа представления от информационно-вычислительной платформы; модель представления знаний должна поддерживать операции, необходимые для представления контекста и управления им.

В рамках задачи *сбора контекстной информации* контекст должен предоставлять релевантную, доступную и достоверную информацию для решения конкретной задачи; включаемая информация должна содержать непосредственно получаемые данные, историю получения этих данных и знания, которые на данный момент известны взаимодействующим компонентам.

В части *интерпретации* контекста управление контекстом должно предусмотреть механизмы, применение которых позволяет использовать собранную информацию соответствующим образом при конкретных условиях для конкретных целей. Данные механизмы включают в себя механизмы, позволяющие: 1) логически выводить новый контекст из имеющихся; 2) повторно использовать контекст с помощью применения контекстов более высоких уровней абстракции, их интеграции и конкретизации для данных условий и задач; 3) получать контекст более высокого уровня абстракции из данного рассматриваемого контекста; 4) разбивать контекст на составляющие его логически связанные внутренне согласованные контексты.

Контексты должны *храниться* в виде, позволяющем постоянное, независимое от текущих задач накопление контекстной информации, хранение истории получения и изменений контекстов и хранение решений, принятых в ситуациях, описываемых контекстами.

В части *распространения* контекстной информации последняя должна быть предоставлена всем заинтересованным компонентам систем поддержки принятия решений в адекватно понимаемом всеми компонентами виде.

Принимая во внимание *динамичный* характер контекстной информации, управление контекстом должно предусматривать методы, позволяющие модифицировать, обновлять и распространять контекстную информацию в масштабе времени близком к времени изменения ситуаций (иногда в масштабе времени близком к реальному времени).

На основании сформулированных требований к управлению контекстом разработана методология управления контекстом для интеллектуальной поддержки принятия решений в открытой информационной среде.

2.2. Методология управления контекстом

Методология управления контекстом ориентирована на формирование модели задачи пользователя (или текущей ситуации), объединяющей в себе контекстно-зависимые информацию и знания, с последующей интерпретацией данной задачи как динамической задачи удовлетворения ограничений. Данная методология рассматривается в рамках концепции интеллектуальной поддержки принятия решений (рис. 1).

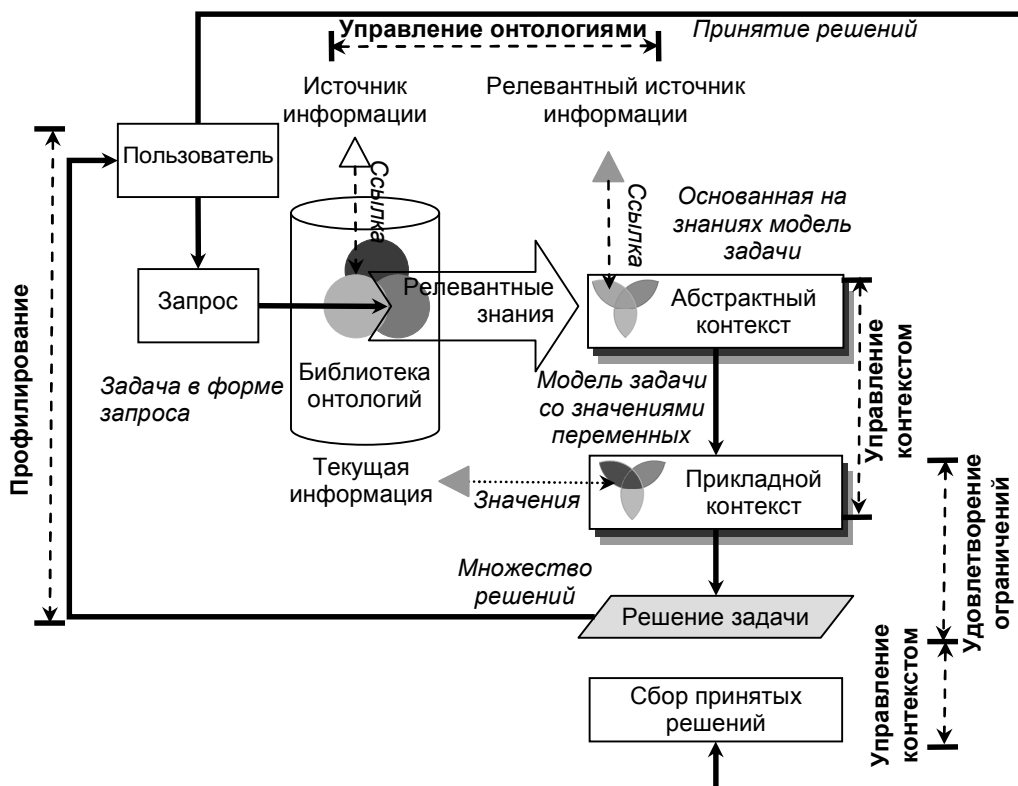


Рис. 1. Концепция интеллектуальной поддержки принятия решений.

Контекст задачи описывается контекстами двух типов: абстрактным и прикладным. *Абстрактный контекст* представляет собой модель задачи, построенную на основании интеграции знаний проблемной области, релевантных для данной задачи. *Прикладной контекст* является экземпляром абстрактного контекста с приписанными текущими значениями переменным, входящих в модель задачи. Значения переменных подаются источниками информации в динамическом режиме.

В зависимости от того, на каком этапе находится процесс формирования решений, в подходе используются различные технологии. На этапе взаимодействия с пользователем и для организации персонализированной поддержки принятия решений используется технология *профилирования*. Для определения релевантных знаний и их интеграции в абстрактный контекст используется технология *управления онтологиями*. Определение релевантной для задачи пользователя информации, ее интеграция в контекст, получение данных от источников информации, присваивание значений переменным, сбор и анализ принятых решений, основанный на управлении версиями контекстов, являются функциями технологии *управления контекстом*. Задача пользователя интерпретируется как задача удовлетворения ограничений, для ее решения используется технология *удовлетворения ограничений*.

В качестве средства представления знаний и описания семантики компонентов открытой информационной среды используется онтологическая модель. Компоненты среды делятся на внешние источники информации и знаний (пользователь, запрос пользователя, базы данных и знаний, сенсоры и другие) и внутренние источники знаний (онтологии проблемной области, абстрактный и прикладной контексты). Между онтологиями проблемной области и внешними источниками устанавливаются связи, показывающие, от какого источника могут подаваться значения свойствам объекта (значения переменных в модель задачи) (рис. 2).

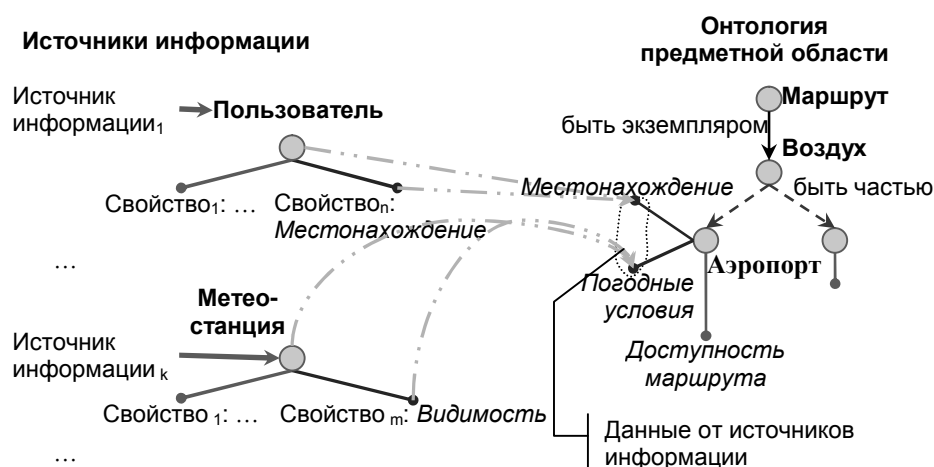


Рис. 2. Связывание онтологии и источников информации.

Отправной точкой этапа принятия решений является запрос пользователя, содержащий формулировку задачи в заданном пользователем виде (как правило, в виде запроса, описываемого как «свободный текст»). После синтаксического разбора запроса ищутся соответствия между словарем запроса и словарями онтологий из библиотеки. На основании найденных соответствий из данных онтологий формируются их срезы [19], содержащие релевантные запросу знания. Данные срезы интегрируются в общий срез. Интеграция знаний, основанная на онтологической модели, позволяет использовать средства логического вывода для проверки получаемого среза на непротиворечивость [20, 21]. Проверенный на непротиворечивость срез представляет собой абстрактный контекст задачи пользователя.

На основании связей с источниками информации, заранее установленных для атрибутов онтологий, абстрактный контекст также содержит ссылки на ис-

точники информации. С абстрактным контекстом оказываются связанными только те источники, которые были связаны с элементами онтологической модели, включенными в абстрактный контекст в качестве описывающих релевантные знания. Прикладной контекст формируется из абстрактного контекста посредством получения от источников знаний и назначения значений свойствам объектов, содержащихся в абстрактном контексте. Прикладной контекст представляет собой модель задачи пользователя с полностью или частично определенными значениями свойств объектов.

Процесс принятия решений является сложным процессом, где на окончательное решение влияет множество факторов. Для того чтобы наиболее естественно учесть все эти факторы, используется механизм объектно-ориентированных сетей ограничений (ООСО), в соответствии с которым задача моделируется множеством ограничений [19].

2.3. Объектно-ориентированная сеть ограничений как модель представления знаний

Знания в рамках формализма ООСО описываются следующими множествами:

- множество классов объектов;
- множество атрибутов классов;
- множество доменов атрибутов;
- множество ограничений, состоящее из множеств:
 - отношений (класс, атрибут, домен), используемых для описания принадлежности атрибутов классам и ограничений на домены данных атрибутов;
 - таксономических (быть экземпляром) и иерархических (быть частью) отношений, используемых для моделирования таксономии и иерархии классов соответственно;
 - отношений совместимости классов, используемых для задания факта, когда класс не может быть частью другого класса;
 - ассоциативных отношения, используемых для моделирования любых отношений, не вошедших в перечисленные, и для импортируемых отношений, которые не могут быть представлены средствами данного формализма;
 - ограничений на множество значений атрибутов, используемых для задания, сколько значений данный атрибут может принимать;
 - функциональных ограничений, используемых для моделирования функций, описывающих взаимосвязь атрибутов.

Соответствия между примитивами онтологической модели, формализмом ООСО и описанием задачи удовлетворения ограничений показаны в табл. 1.

Формализм ООСО используется как внутренний формализм описания знаний, как формализм для представления всех компонентов открытой информационной среды и как формализм для описания сообщений агентов. Он совместим с языками описания знаний DAML+OIL и OWL используемых в рамках технологии Semantic Web [22], и удобно описывается средствами языка XML (Extensible Markup Language), используемого в качестве базового языка для разработки распределенных приложений в рамках технологии W3C (World Wide Web Consortium) [23].

Соответствие между онтологической моделью, формализмом ООСО и задачей удовлетворения ограничений

Онтологическая модель	ООСО	Задача удовлетворения ограничений
Класс	Объект (класс)	Переменная
Атрибут	Переменная	
Область допустимых значений атрибута	Домен	Домен
Отношения и аксиомы	Ограничения	Ограничения

3. Многоагентная архитектура технологии правления контекстом

Набор компонент открытой информационной среды определяет ее распределенную архитектуру. Так как знания, вовлеченные в процесс управления контекстом, находятся в разнородных распределенных ресурсах для организации взаимодействия между данными ресурсами и их содержимым выбрана *технология интеллектуальных агентов*. Интеллектуальные агенты широко используются в системах управления знаниями, ориентированных на работу с распределенными и разнородными источниками знаний.

Архитектуру разработанной технологии предложено реализовывать на основе модели управления агентами FIPA (Foundation for Intelligent Physical Agents) [24]. При этом технология интеллектуальных агентов используется для организации взаимодействия между компонентами открытой информационной среды и реализации операций по управлению контекстом. Для этих целей определены агенты двух типов: технологические и проблемно-ориентированные:

Технологические агенты выбраны в соответствии со спецификацией FIPA [24]:

- посредник. Данный агент выполняет функции осуществления взаимодействия между агентами, входящими в многоагентную архитектуру.
- маршрутизатор (в спецификации FIPA — Directory Facilitator): агент, предоставляющий другим агентам справочную службу «желтых страниц». Для рассматриваемой платформы функции данного агента расширены функциями агента управления многоагентной платформой (в спецификации FIPA — Agent Management System).

Набор проблемно-ориентированных агентов определен в соответствии с целями и задачами управления контекстом. Данный набор включает в себя как типовых агентов, специфицированных для многоагентных платформ [25], так и агентов, специально разработанных для решения рассматриваемых задач. Функции типовых агентов расширены функциями, требуемыми в многоагентной платформе, предназначенной для управления контекстом. Из типовых агентов в многоагентную платформу включены:

- агент пользователя. Данный агент осуществляет взаимодействие с пользователями системы.
- интерфейсный агент. Данный агент осуществляет взаимодействие с источниками знаний.

Дополнительно для задач управления контекстом специфицирован следующий набор проблемно-ориентированных агентов:

- агент-помощник эксперта. К функциям данного агента относятся функции по поддержке сценариев работы экспертов проблемной области и инжене-

ров по знаниям в рамках технологии интеллектуальной поддержки принятия решений;

- транслирующий агент. Функциями данного агента являются функции по переводу словарей и по преобразованию форматов представления знаний;
- агент управления контекстом. К функциям данного агента относятся функции по созданию абстрактного и прикладного контекстов;
- агент-конфигуратор. Данный агент отвечает за эффективность использования сети источников знаний;
- агент управления онтологиями. К функциям данного агента относятся функции по выполнению операций над онтологиями;
- агент мониторинга. Данный агент предоставляет оперативную информацию о состоянии источников знаний и о содержимом данных источников.

Многоагентная архитектура технологии управления контекстом представлена на рис. 3. В соответствии с данной архитектурой разработаны модели агентов, специфицированы наборы, выполняемых ими процедур и функций (рис. 4).

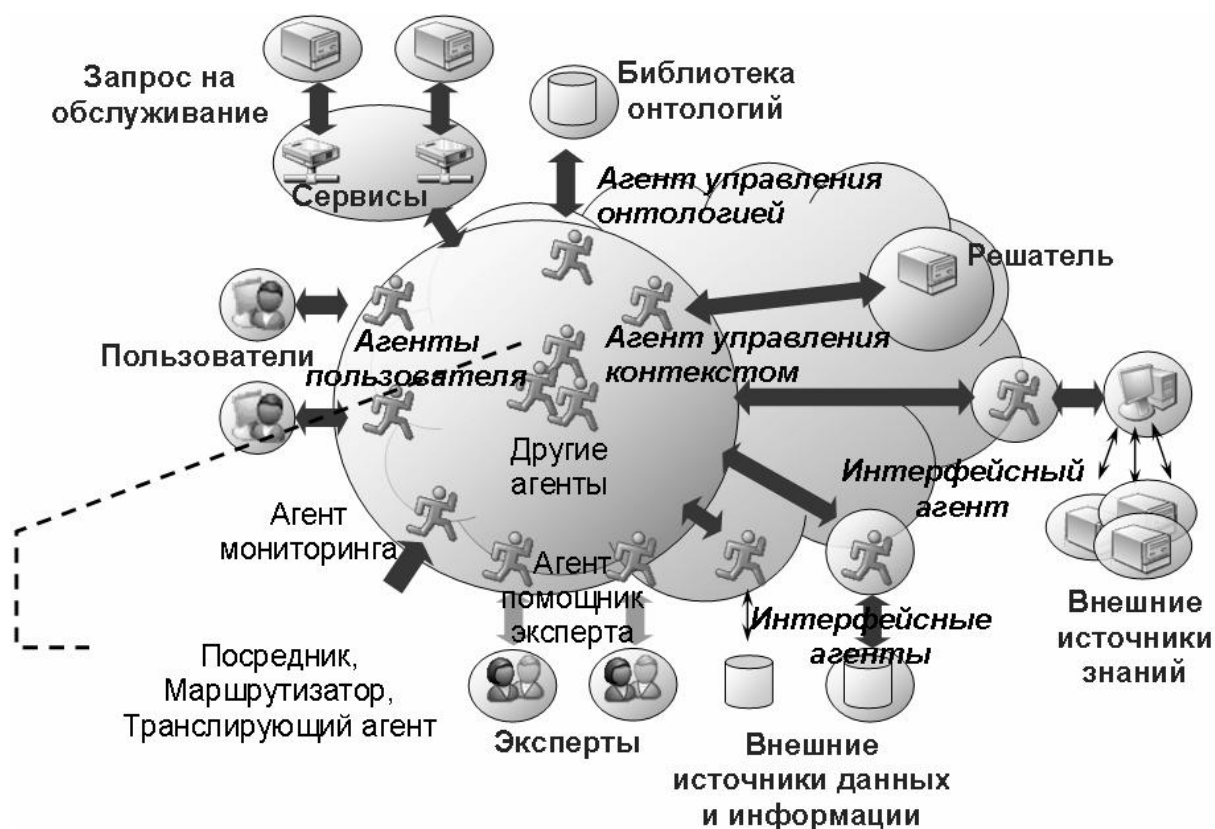


Рис. 3. Многоагентная архитектура.

В рамках концепции интеллектуальной поддержки принятия решений предполагается два вида взаимодействия между агентами: равноправное и опосредованное. В табл. 2 приведена матрица взаимодействия агентов, иллюстрирующая типы взаимодействия между агентами технологии управления контекстом.

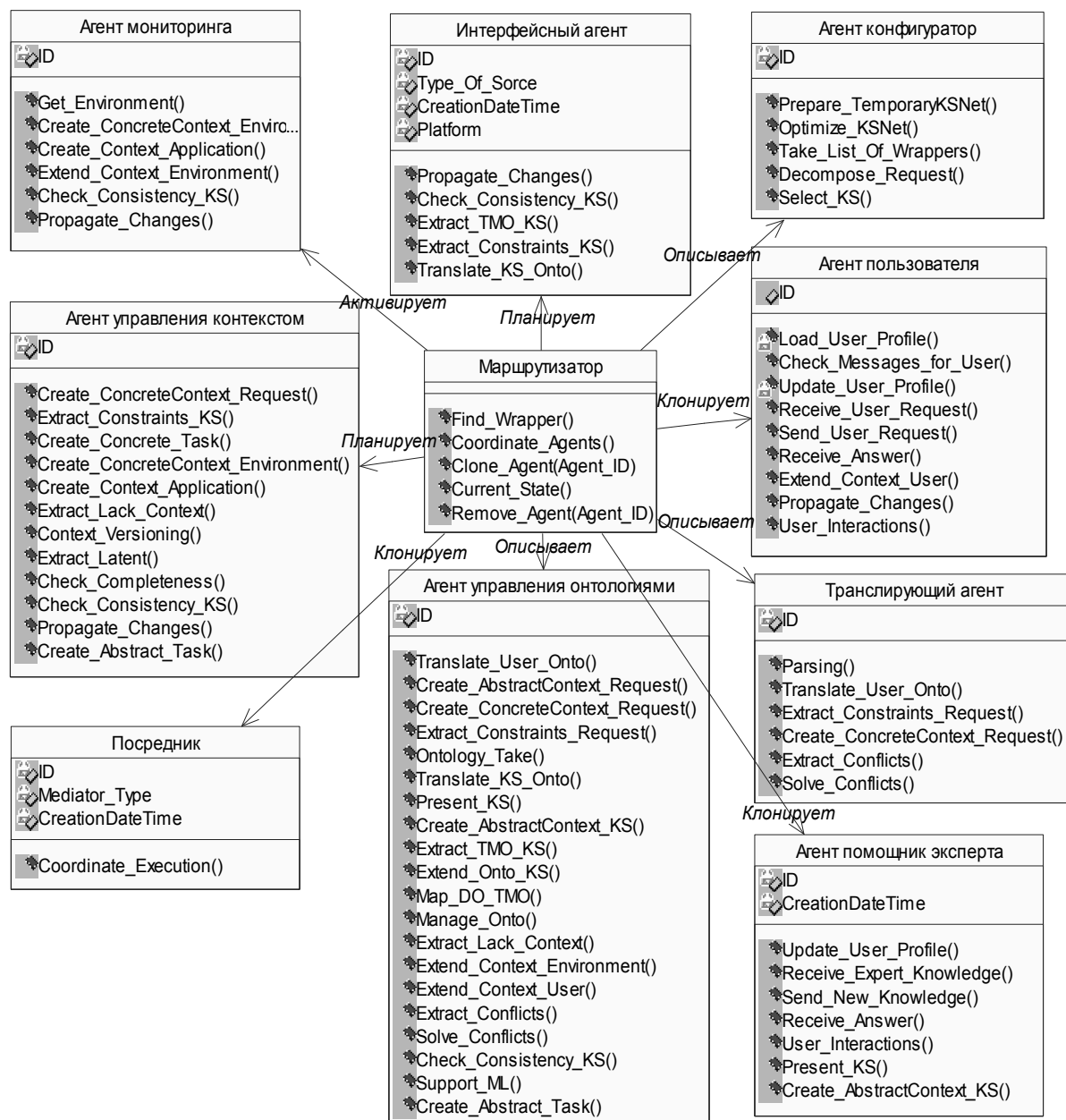


Рис. 4. Агенты технологии управления контекстом: основные функции и свойства.

Для осуществления взаимодействия агентов в многоагентной платформе определены два состояния, в которых данные агенты могут находиться: 1) «Работающий», когда агент выполняет один или несколько сценариев и 2) «Свободный», когда агент ожидает получения сообщений от других агентов. В начале работы системы только агенты мониторинга и маршрутизатор находятся в состоянии «Работающий», выполняя сенсорные функции. Остальные агенты находятся в состоянии «Свободный». Получив сообщение от агентов среды, они переходят в состояние «Работающий», выполняют свои функции в соответствии с заданным сценарием и снова переходят в состояние «Свободный». Агенты-помощники эксперта и посредники изначально в системе не существуют. Их количество зависит от числа зарегистрированных экспертов и выполняемых заданий. Поэтому, их сущности клонируются и разрушаются по мере необходимости. После того, как сущность указанных агентов клонирована, они находятся в состоянии «Работающий».

3.1. Технологические агенты

Посредник. Посредник сопровождает задание по этапам его выполнения с момента поступления до завершения. Он предназначен для распределения работ (информации) между агентами, включенными в многоагентную архитектуру. Имеет дополнительную характеристику — тип выполняемого задания. Возможны два типа заданий: сопровождение обработки запроса пользователя и сопровождение работы экспертной группы (инженеров по знаниям, онтологиям и экспертов).

Новый посредник клонируется маршрутизатором при возникновении нового задания. Сразу после клонирования посредник ожидает взаимодействия с агентами среды. При отсутствии обращения в течение заданного интервала времени со стороны данных агентов, посредник удаляется из системы. После того, как задание, которое сопровождал посредник, выполнено, сущность данного агента уничтожается.

Кроме осуществления взаимодействия между агентами в ходе сопровождения задания, посредник выполняет контроль ошибок, системных сбоев и временных задержек при получении ответов от других агентов, протоколирование ошибок и сбоев.

Маршрутизатор. Задачами данного агента является поддержка списка агентов, функционирующих в данный момент в среде, и предоставление оперативной информации об агентах тем агентам, которые имеют на данную информацию права (авторизованным агентам). Маршрутизатору добавлены функции управления операциями над агентами (создание и удаление агентов, отслеживание появления агентов в среде и выхода агентов из среды) и предоставления описания текущего состояния многоагентной среды. Маршрутизатор присутствует в среде.

Маршрутизатор хранит список зарегистрированных в системе агентов и предоставляемых ими услуг, отслеживает их состояние («работающий» или «свободный»), ведет расписание работы агентов.

После начала работы все агенты сообщества, кроме маршрутизатора, находятся в состоянии «свободный». Маршрутизатор меняет состояние агента мониторинга на «работающий». Далее основными функциями маршрутизатора являются клонирование и назначение заданий агентам пользователей, посредникам и агентам-помощникам экспертов в зависимости от события, возникающего в системе.

Модель поведения агента маршрутизатора определяется тремя типами полученных им сообщений: 1) запрос на предоставление агента для выполнения задания; 2) сообщение от агентов, закончивших работу над заданиями, что они свободны; 3) запрос с просьбой предоставить информацию об агентах среды. Получив сообщение первого вида, маршрутизатор создает новую сущность требуемого агента, вносит изменения в свой репозиторий, изменяет состояние созданного агента на «работающий» и посылает его идентификатор агенту, приславшему запрос. В случае сбоя маршрутизатор возвращает сообщение об ошибке. Получив сообщение второго вида, маршрутизатор меняет в своем репозитории информацию о состоянии агента на «свободный» и уничтожает его сущность. Получив сообщение третьего вида, маршрутизатор ищет требуемую информацию в репозитории и предоставляет ее в ответном сообщении. Сообщения первой и третьей группы являются синхронными, второй группы — асинхронными.

Матрица взаимодействия агентов

Вызываемый \ Вызывающий	Интерфейс-ный	Посредник	Маршрутизатор	Пользователя	Транслирующий	Помощник эксперта	Конфигуратор	Управления контекстом	Мониторинга	Управления онтологиями
Интерфейсный		Р			Р				Р	Р
Посредник	О/Р		Р	Р	Р	Р	Р	Р	Р	
Маршрутизатор		Р		Р		Р			Р	
Пользователя		О/Р	Р							
Транслирующий		Р								Р
Помощник эксперта		О/Р	Р							
Конфигуратор	О/Р	Р	Р					О/Р	Р	
Управления контекстом		Р								
Мониторинга	О/Р	О/Р	Р				Р			
Управления онтологиями	Р				Р				Р	

Примечание. Р — равноправное, О — опосредованное.

3.2. Проблемно-ориентированные агенты

Агент пользователя. Агент пользователя сопровождает ввод запросов в среду, предоставляет подсказки пользователю, передает пользователю сообщения от среды, обсуживает профиль пользователя. Агент пользователя выполняет функции по созданию, модификации, и обновлению профилей пользователей и накоплению знаний о пользователе; по модификации абстрактного и прикладного контекстов пользователя.

Новый агент пользователя клонируется после прохождения пользователем процедуры аутентификации или после завершения работы экспертной группы над запросом, обработка которого не была завершена, Сразу после клонирования он проверяет профиль пользователя на наличие новых сообщений и ожидает заданий пользователя или продолжает обработку запроса с указанного места. При отсутствии в течение заданного интервала времени обращения со стороны пользователя и после окончания работы пользователя со средой, агент удаляется из системы.

В части взаимодействия пользователя со средой поддержки принятия решений разработана *модель профиля пользователя* (рис. 5), учитывающая особенности методологии управления контекстом и позволяющая накапливать знания о пользователе, его интересах и предпочтениях. Модель профиля пользователя описывается в соответствии с рекомендациями для профиля пользователя, разработанными в рамках технологии W3C (World Wide Web Consortium) [23] и задающими структурные элементы профиля пользователя для описания текущего сеанса взаимодействия пользователя со средой. Для учета специфики управления контекстом используемая модель, кроме свойств, рекомендуемых указанными предложениями, дополнена классами «персональные данные», «системные данные» и «накопленная информация». Данные классы содержат подклассы и атрибуты позволяющие описывать свойства пользователя, характерные для разрабатываемой технологии, такие, как место нахождения пользователя, история запросов пользователя, область интересов пользователя. В случае необходимости, в функциях агента пользователя предусмотрены функции, позволяющие расширить данную модель.

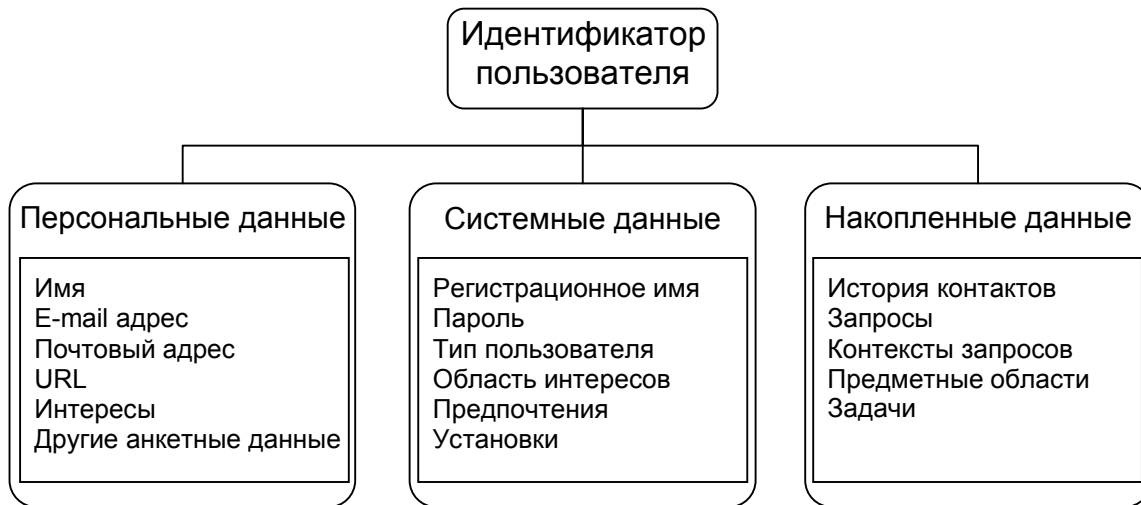


Рис. 5. Структура профиля пользователя.

Интерфейсный агент. Интерфейсный агент выполняет функции по отображению между словарями источника знаний и словарем, используемым для описания онтологий; функции по извлечению ограничений из источника знаний, процедуры по выявлению составляющих онтологий предметных областей и онтологий задач и методов из источников знаний; функцию распространения контекстных изменений в части распространения изменений, касающихся изменений в источниках знаний; функцию по проверке непротиворечивости знаний, предоставляемых источником знаний; процедуру управления версиями контекстов источников знаний.

Интерфейсный агент предназначен для работы с определенными типами источников знаний. Тип источника знаний, с которым данный агент взаимодействует, является характеристикой интерфейсного агента. Может работать как в параллельном, так и в однозадачном режимах. Режим определяется типом источника знаний. Работает согласно расписанию, согласованному с агентом-конфигуратором на этапе переговоров.

В связи с тем что открытая информационная среда имеет динамическую природу с динамически меняющимися источниками знаний, количество интерфейсных агентов не является постоянной величиной. В связи с этим в состав многоагентной платформы могут вводиться новые интерфейсные агенты, а свойства и функции существующих могут меняться. Интерфейсные агенты распределяются по порталам, обеспечивающим доступ к источникам знаний. Для обработки источников знаний, имеющих сходную структуру, но недоступных из одного портала, копии (клоны) существующих интерфейсных агентов располагаются в разных порталах.

Агент-помощник эксперта. Сценарий поведения агента-помощника эксперта связан со сбором результатов работы экспертов (инженеров по знаниям или онтологиям) и их передаче заинтересованным агентам. Новый агент-помощник эксперта клонируется после того, как эксперт проходит процедуру аутентификации. При отсутствии в течение заданного интервала времени обращения со стороны эксперта или окончании работы эксперта, агент удаляется из системы. В части персонифицированной поддержки экспертов используется модель профиля пользователя (рис. 6), где в элементе структуры профиля, описывающего «тип пользователя», заносится соответствующий тип (эксперт

проблемной области, инженер по знаниям, инженер по онтологиям и тому подобное).

Данный агент сопровождает действия экспертов при вводе знаний в библиотеку онтологий; при представлении знаний, предоставляемых внешними источниками, для которых не разработаны средства автоматического импорта, средствами внутреннего формализма; при разрешении конфликтных ситуаций при представлении знаний различными источниками; при устранении противоречий при интеграции знаний; при недостаточности имеющихся в библиотеке онтологий знаний для решения задачи пользователя.

Транслирующий агент. Функциями данного агента являются функции по переводу словарей и по преобразованию форматов представления знаний, а сценарий его поведения связан с 1) обработкой запроса пользователя и 2) источников знаний.

В первом случае транслирующий агент получает текст запроса для проведения предварительной обработки. Он удаляет незначимые слова, проверяет синтаксические ошибки, формирует канонические формы слов, переводит словарь запроса в словарь среды поддержки принятия решений, готовит XML структуру, содержащую информацию о словах запроса. Результаты обработки запроса служат исходными данными для формирования абстрактного контекста (задача формирования абстрактного контекста решается другим агентом). Транслирующий агент получает запрос пользователя и распознает в нем, если это возможно, подзапросы (образцы запросов, сформированные в процессе обработки профилей пользователей). Получив результаты работы среды поддержки принятия решений, транслирующий агент переводит их в представление, понятное пользователю.

Во втором случае транслирующий агент получает запрос на обработку источника знаний и адрес данного источника. Он переводит словарь источника в словарь среды поддержки принятия решений, формат представления знаний в источнике — в формализм объектно-ориентированных сетей ограничений. Передает результаты обработки источника знаний заинтересованному агенту. При получении запроса на перевод внутреннего представления во внешнее, для источников знаний, включенных в картограмму знаний, агент преобразовывает внутренний формализм и словарь в формализм и словарь данного источника знаний. Возможен вариант простого экспорта знаний, представленных в формализме объектно-ориентированных сетей ограничений, во внешний формат, поддерживаемый транслирующим агентом (ТА). В этом случае используемый словарь не переводится.

Одна из задач агента — это выделение параметрической и структурной составляющей в запросе пользователя. Агент пользователя передает ТА запрос пользователя, заданный или на естественном языке, или с помощью шаблонов, которые облегчают процесс распознавания запроса. В результате работы ТА создается XML -сообщение [26], которое содержит список ключевых слов и их значений (имена классов и атрибутов, хранящихся в онтологии) — структурная составляющая, а также список ограничений на значения атрибутов, где значения — это параметрическая составляющая. Это XML сообщение затем передается агенту управления онтологиями для создания предварительной онтологии запроса, которая входит в область интересов пользователя (профиль пользователя).

Для каждого ключевого слова указывается список синонимов, например, синоним «лазарет» (infirmary) для слова «больница» (hospital). Синонимы нуж-

ны для того, чтобы расширить возможности поиска онтологий при работе с библиотекой онтологий. Для поиска синонимов в английском языке была использована свободно распространяемая система WordNet [27, 28, 29], которая применялась во многих проектах в области информационного поиска [30, 31, 32, 29].

Для ключевых слов в XML сообщении (тег `<Word>` на рис. 6) может быть указан тип слова (тег `<Type>`). (Тег — это именованная конструкция языка разметки XML [26].) После разбора транслирующим агентом запроса пользователя и на его основе выполняется поиск онтологий, соответствующих запросу. Указание на этом этапе типа слова позволяет структурировать запрос пользователя и облегчить последующий поиск в библиотеке онтологий. В системе «Интеграция» онтология описывается с помощью классов объектов, классов задач и методов, атрибутов, ограничений, поэтому ТА поддерживает и распознает четыре типа слов: класс (Class), задача (Action), атрибут с заданным ограничением на значение (Attribute) и нераспознанное слово (Unrecognized). Тип задача (Action) указывает на класс из онтологии задач и методов. Если слово из запроса пользователя не отнесено ни к одному из указанных типов, то тип слова не указывается, то есть тег `<Type>` отсутствует (например, ключевое слово “mobile” на рис. 6).

При определении типа слова ТА использует несколько синтаксических правил (для предложений на английском языке), с помощью которых слово относится к одному из типов:

- **Класс**, если слово стоит после артикля (“a” или “an”) и до союза (“with” или “where”), например, словосочетание “mobile hospital” в контексте “a mobile hospital with price”.
- **Задача**, если слово стоит после приинфинитивной частицы “to” и до артикля “a” или “an”, например, слово “build” в контексте “to build a hospital”.
- **Атрибут** — это конструкция, в которой выделяют три составляющие: (i) имя атрибута (тег `<Word>`), (ii) тип ограничения (тег `<Operator>`) и (iii) задаваемое значение (тег `<Value>`). Тип ограничения в прототипе задается следующими математическими знаками: `<`, `>`, `=`, `<=`, `>=`.
- **Нераспознанное слово**, если слово не принадлежит ни одному из вышеприведенных типов и отсутствует в словаре WordNet.

Алгоритм обработки запроса пользователя транслирующим агентом (расширение — поиск синонимов и структуризация запроса — типизация слов):

- 1) Чтение файла с запросом пользователя.
- 2) Разбор сообщения: извлечение ограничений (тип атрибут).
- 3) Разбор сообщения: извлечение слов, относимых к типу «класс» и «задача» и тех слов, которые не относятся к какому-либо типу (извлечение только уникальных слов, если слово повторяется, то его пропускают).
- 4) Поиск синонимов для каждого уникального слова с помощью WordNet (перед добавлением очередного слова в результирующий набор слов проверяется отсутствие данного слова в наборе).
- 5) Сохранение в файл извлеченных из запроса пользователя ключевых слов.

```

<?xml version="1.0" standalone="yes"?>
<!--KSNet 2002 SPIIRAS-->
<Root_Element><!--В таких скобках пишут xml-комментарии -->
  <KSWord> <!--Начало блока информации для слова «build» -->
    <Word>build</Word><!-- Определяем слово «build»-->
    <Type>Action</Type> <!--Тип слова «build» -->
    <Synonym>construct</Synonym> <!--Синоним слова «build» -->
    <Synonym>make</Synonym> <!--Синоним слова «build» -->
  </KSWord> <!--Конец блока информации для слова «build» -->
  <KSWord><!--Для фразы «mobile hospital» не найдено синонимов-->
    <Word>mobile hospital</Word>
    <Type>Class</Type> <!--Тип фразы «mobile hospital»-->
  </KSWord>
  <KSWord>
    <Word>mobile</Word>
  </KSWord>
  <KSWord>
    <Word>hospital</Word>
    <Synonym>infirmary</Synonym><!--Синоним слова «hospital»-->
  </KSWord>
  <KSWord>
    <Word>cost</Word> <!--Имя переменной «cost» -->
    <Type>Attribute</Type><!--Тип конструкции «price<=1000»-->
    <Operator>&lt;=&/Operator><!--Тип ограничения «<=» -->
    <Value>10000</Value> <!--Значение ограничения «1000» -->
    <Synonym>price</Synonym><!--Синоним слова «cost» -->
  </KSWord>
</Root_Element>

```

Рис. 6. Пример успешно распознанного запроса в виде XML-сообщения, содержащего список распознанных слов с указанием типов и перечислением синонимов.

Запрос пользователя разбирается на основе использования регулярных выражений языка Perl [34]. Регулярные выражения — это специальные шаблоны, которые часто используются при обработке текстов. Такие известные утилиты Unix, как: *grep*, *sed* и *awk*, а также язык программирования *perl* — активно используют регулярные выражения. Динамическая библиотека свободно распространяемого продукта компании ActiveState [35] обеспечивает возможность работы с регулярными выражениями, соответствующими семантике языка Perl. После подключения этой библиотеки к проекту MS Visual Studio доступ к ее функциям становится возможным из C++ кода. На рис. 6 представлен пример XML сообщения для распознанного запроса на естественном языке: “*To build a mobile hospital with price <= 10000*”.

Если запрос не был распознан или распознан частично, то для решения этой проблемы формируется группа экспертов. Эксперты проверяют нераспознанные слова из запроса пользователя, выделяют структурную и параметрическую составляющие запроса. Результаты работы экспертов передаются далее агенту управления онтологиями.

Агент управления контекстом. Задачи агента управления контекстом состоят в выполнении операций по созданию контекстов и по управлению контекстом. Для данного агента характерны функции по формированию прикладных контекстов запроса пользователя, источника знаний и окружающей среды; функции по формированию абстрактного и прикладного контекстов задачи; функция, выявляющая компонент среды, знания от которого требуются для решения задачи на основании анализа системы ограничений контекста задачи, связанных с контекстами пользователя, источников знаний и окружающей среды; процедура по распространению контекстных изменений; процедура проверки полноты описания запроса пользователя контекстом задачи; набор процедур

по управлению версиями контекстов; процедура выявления скрытых знаний; функция по проверке непротиворечивости информации, предоставляемой источниками знаний.

Сценарий взаимодействия агента управления онтологиями и агента управления контекстом на предмет создания абстрактного контекста задачи. Агенты начинают работу в соответствии с данным сценарием после получения агентом управления онтологиями сообщения, содержащего результаты отображения словаря запроса пользователя в словарь, поддерживаемый онтологиями. *Агент управления онтологиями* формирует срезы из онтологий и представлений компонентов системы, содержащие релевантные для поступившего запроса знания. *Агент управления контекстом* объединяет сформированные срезы, проверяет результат объединения на внутреннюю согласованность и, в случае положительного результата, объединенные срезы рассматриваются как абстрактный контекст задачи. В случае несогласованного среза, вызываются функции по разрешению конфликтных ситуаций.

Сценарий взаимодействия агента управления онтологиями и агента управления контекстом на предмет анализа абстрактного контекста задачи с целью выявления контекста, содержащего недостающие для задачи информацию / знания. Агенты начинают работу в соответствии с данным сценарием после получения агентом управления онтологиями сообщения от агента управления контекстом о недостаточности абстрактного контекста задачи для решения задачи, представленной в запросе пользователя. Данное сообщение содержит список терминов запроса, не представленных в контексте, и (или) набор входных параметров методов, для которых не установлено соответствие с атрибутами классов онтологии предметной области, т.е. не определено, значения каких атрибутов будут служить входными параметрами при решении задачи. *Агент управления онтологиями* 1) на основании анализа иерархии классов и правил наследования атрибутов и отношений определяет, значения атрибутов какого класса могут служить входными параметрами метода; 2) анализирует ссылки на компоненты среды, связанные с данным классом; 3) делает вывод о том, какой компонент отвечает за недостающие знания. Информация о данном компоненте передается агенту управления контекстом, он просматривает архив контекстов на предмет решения подобных задач и делает вывод, какие знания в сохраненных версиях контекстов данного компонента могут компенсировать недостаточность контекста задачи. Если агент управления контекстом не смог выявить недостающие знания, то агент управления онтологиями взаимодействует с компонентом системы (через посредника) с целью уточнения контекста данного компонента.

Сценарий взаимодействия интерфейсного агента и агента управления контекстом при формировании прикладного контекста источника знаний. Агенты начинают работу в соответствии с данным сценарием после получения интерфейсным агентом сообщения, содержащего информацию о том, какие значения предоставляет данный источник знаний в прикладной контекст задачи. *Интерфейсный агент* извлекает запрашиваемые значения и передает их агенту управления контекстом в виде сообщения, содержащим пары, описывающие соответствие между элементами абстрактного контекста и их значениями в источнике знаний. *Агент управления контекстом* приписывает полученные значения включенным в абстрактный контекст атрибутам классов.

Сценарий взаимодействия агента управления онтологиями, агента пользователя и агента управления контекстом на этапе поддержки контек-

стов пользователя. Агенты начинают свою работу в соответствии с данным сценарием после получения агентом пользователя от агента управления онтологиями сообщения, что для контекста задачи требуется дополнительная информация о пользователе. Данная информация выявляется в процессе взаимодействия агента пользователя с пользователем. Агент пользователя расширяет профиль пользователя и абстрактный контекст пользователя, используемый в данной задаче и посылает 1) сообщение агенту управления онтологиями, содержащее информацию о том, каким образом был модифицирован абстрактный контекст пользователя; 2) сообщение агенту управления контекстом о том, каким элементам абстрактного контекста какие значения должны быть присвоены в прикладном контексте рассматриваемой задачи. Агент управления онтологиями вносит соответствующие изменения в онтологию. Агент управления контекстом приписывает полученные значения указанным элементам контекста.

Агент-конфигуратор. *Агент-конфигуратор* отвечает за эффективность использования сети источников знаний. Его функциями являются декомпозиция запроса пользователя на подзапросы; поиск по картограмме знаний источников знаний, из которых будут извлекаться данные, информация и знания для обработки запроса пользователя, и интерфейсных агентов, обслуживающих данные источники знаний; построение в процессе обработки запроса пользователя временной конфигурации сети источников знаний.

Данный агент получает запрос на определение источников знаний, требующихся для решения описанной в абстрактном контексте задачи пользователя. На основании анализа классов онтологий, включенных в абстрактный контекст, агент в соответствии со стратегиями, определенными для типовых запросов, формирует XML структуру, содержащую информацию об источниках знаний, которые могут быть использованы в данном контексте. Результатом работы агента является сеть источников знаний, которая строится с использованием генетического алгоритма, и характеристики источников знаний, включенных в сеть (время, когда источник доступен, его стоимость и тому подобные).

Агент управления онтологиями. *Задачи агента управления онтологиями* связаны с операциями над онтологиями. Для данного агента определен набор функций по отображению словарей запросов пользователей в словарь, используемый для описания онтологий; набор процедур по формированию абстрактных контекстов запроса и источника знаний; процедура определения релевантных по отношению к текущей задаче знаний; процедура формирования срезов онтологий, содержащих релевантные для данной задачи знания; процедура формирования абстрактного контекста задачи; процедура формирования прикладного контекста запроса; набор функций по модификации абстрактного и прикладного контекстов пользователя и окружающей среды; набор функций по модификации онтологий, хранящихся в библиотеке, и по созданию новых онтологий; набор процедур по управлению версиями онтологий; набор процедур по установке соответствий между онтологиями предметной области и задач и методов; функция, выявляющая компонент среды, знания от которого требуются для решения задачи на основании анализа системы ограничений контекста задачи, относящейся к представлению знаний проблемной области; процедуры по выявлению и разрешению семантических конфликтов в контекстах компонентов среды; функция по выявлению знаний, составляющих онтологию пред-

метных областей и задач и методов из источников знаний; процедуры по поддержке библиотеки методов.

Агент мониторинга. *Агент мониторинга* служит для проверки состояния источников знаний и их содержимого, обеспечения взаимодействия с внешними приложениями открытой информационной среды и взаимодействия с окружающей средой. Для рассматриваемого агента определены сенсорные функции получения данных от окружающей среды; функции проверки непротиворечивости абстрактного контекста окружающей среды; передачи полученных данных в прикладной контекст окружающей среды и в прикладной контекст задачи; управления контекстами окружающей среды; модификации абстрактного и прикладного контекстов окружающей среды; диагностирования картограммы знаний; предоставления оперативной информации о состоянии источников знаний и о содержимом данных источников посредством мониторинга данных источников на предмет их доступности и изменений в их содержимом.

Модель поведения агента *мониторинга* включает в себя сценарии по 1) обработке картограммы знаний, 2) сопровождению работы экспертной группы, 3) моделированию окружающей среды.

В первом сценарии агент мониторинга запрашивает у маршрутизатора список зарегистрированных интерфейсных агентов и, проведя сравнение с поддерживаемой картограммой знаний, определяет, какие интерфейсные агенты отсутствуют. Посылает широковещательное синхронное сообщение зарегистрированным интерфейсным агентам и синхронное сообщение агенту управления онтологиями. По ответам, полученным от этих агентов, он определяет, какие источники знаний относительно каких элементов абстрактного контекста должны быть проиндексированы. Источники знаний, обслуживаемые интерфейсными агентами, от которых ответ не пришел или которые не зарегистрированы в системе, считаются недоступными. В картограмме знаний уничтожаются ссылки на данные источники. Затем агент мониторинга формирует список интерфейсных агентов, которые должны выполнить предварительную обработку источников знаний, и отправляет им широковещательное сообщение. Производит обновление картограммы знаний.

Получив задание на сопровождение работы экспертной группы (*второй сценарий*), агент мониторинга подготавливает необходимые данные и посылает сообщения посреднику для агентов-помощников экспертов. Набор данных зависит от задачи, которую необходимо решить, это могут быть характеристики источников знаний, по которым будет осуществляться ранжирование данных источников, или текст запроса пользователя, который по каким-то причинам не был обработан. При этом в профили пользователей добавляются сообщения о заданиях, которые должны быть выполнены.

Окружающая среда моделируется (*сценарий 3*) набором данных. Агент мониторинга получает сообщение от агента посредника о том, какие данные требуются для решения задачи пользователя. Снимает при помощи сенсорных функций требуемые значения, при необходимости, преобразовывает данные значения в используемые в задаче единицы измерения этих значений. Проверяет соответствие диапазонов передаваемых значений доменам атрибутов классов, входящих в абстрактный контекст задачи и запрашивающих эти значения. Передает значения в прикладной контекст окружающей среды и в прикладной контекст задачи.

4. Общий сценарий и исследовательский прототип многоагентной технологии управления контекстом

Общий сценарий решения задачи управления контекстом для интеллектуальной поддержки принятия решений на основе многоагентной технологии включает в себя следующие шаги:

- 1) *агент управления контекстом* формирует абстрактный и прикладной контекст приложения;
- 2) пользователь при помощи *агента пользователя* вводит запрос;
- 3) *агент пользователя* запрашивает у *маршрутизатора* идентификатор *посредника*, который будет сопровождать данное задание по обработке запроса пользователя. *Маршрутизатор* клонирует нового *посредника* и назначает ему задание;
- 4) *посредник* получает запрос пользователя от *агента пользователя*;
- 5) *транслирующий агент*, взаимодействуя с *агентом управления онтологиями*, распознает словарь запроса пользователя и возвращает переведенный запрос *посреднику*;
- 6) *агент управления онтологиями* формирует абстрактный контекст запроса;
- 7) *агент управления онтологиями* формирует срезы онтологий, извлекает из запроса дополнительные ограничения, не вошедшие в абстрактный контекст запроса;
- 8) *агент управления контекстом* формирует прикладной контекст запроса;
- 9) *агент управления контекстом* формирует абстрактный контекст задачи;
- 10) *агент управления контекстом* 1) проверяет полноту описания абстрактным контекстом задачи абстрактного контекста запроса пользователя, 2) расширяет абстрактный и прикладной контекст пользователя через взаимодействие с *агентом пользователя*, 3) выявляет из контекста задачи контексты объектов, к которым относится недостающая информация;
- 11) *агент-конфигуратор* определяет источники знаний, соответствующие запрашиваемой контекстом задачи информации, и ведет переговоры с *интерфейсными агентами* на предмет значений атрибутов, вошедших в абстрактный контекст источника знаний;
- 12) *агент управления онтологиями* формирует абстрактный контекст источника знаний;
- 13) *агент мониторинга* определяет данные о состоянии окружающей среды;
- 14) *агент управления контекстом* формирует прикладной контекст окружающей среды;
- 15) *агент управления контекстом* формирует прикладной контекст задачи;
- 16) *агент управления онтологиями* выявляет составляющие онтологии предметной области и задач и методов из источников знаний;
- 17) *интерфейсные агенты* получают ограничения из источника знаний, формируют прикладной контекст источника знаний;
- 18) *посредник* передает результат *транслирующему агенту*. Транслирующий агент переводит решение в представление, понятное пользователю;
- 19) *посредник* передает полученный результат агенту пользователя;
- 20) *агент пользователя* возвращает результат пользователю.

В исследовательском прототипе используется распределенная много-агентная архитектура. Агенты могут располагаться на различных компьютерах, называемых «порталами». Один из компьютеров (названный «сервером»), помимо функции локального размещения агентов, выполняет функцию координации всех других порталов, связанных с ним по протоколу TCP/IP. Каждый портал имеет IP адрес и хранит набор зарегистрированных агентов. Сервер, дополнительно к этому, содержит список IP адресов всех порталов. Для уменьшения сетевого трафика при взаимодействии агентов, для максимального использования мощностей серверного компьютера и, в соответствии с разработанной архитектурой использования общей директории работы, большая часть агентов располагается на сервере. Они могут сохранять промежуточные результаты во временных файлах, а затем читать и модифицировать их содержимое. Используя разработанный механизм формирования имен файлов, каждый агент знает, какой именно файл ему следует использовать для решения определенной задачи. Интерфейсные агенты распределяются по порталам, обеспечивающим доступ к источникам знаний.

Сценарии работы отдельных агентов описаны в виде конечных автоматов, представленных набором состояний. Состоянием начала работы может быть сообщение, полученное от какого-либо агента, или возникновение внешнего события в среде поддержки принятия решений, обнаруженное сенсорными функциями агентов. В каждом состоянии автомата выполняется определенный набор функций, по результатам работы которых происходит переход в новое состояние.

Механизмы формирования и пересылки сообщений, запуска и остановки работы сообщества агентов, реализации конечных автоматов работы агентов обеспечиваются средствами системы MAS DK [36], разработанной в СПИИРАН. Обмен сообщениями в данной системе осуществляется по протоколу KQML, система предоставляет механизмы для отслеживания ошибок в сценариях работы агентов и позволяет быстрый анализ ошибок, присваивая сообщениям разные уровни значимости. Механизмы отслеживания системных сбоев при работе агентов с помощью кодов ошибок и отладочных сообщений, сохраняющихся в журнале работы системы, были разработаны в рамках функций агентов.

MAS DK сохраняет информацию о всех взаимодействиях внутри сообщества агентов в базе данных, реализованной в СУБД Access. Это делает возможным производить анализ взаимодействия агентов встроенными в систему средствами.

Ниже приведен фрагмент сценария, связанный с формированием абстрактного и прикладного контекстов, в рамках многоагентной среды «Интеграция» на примере формирования мобильного госпиталя в ситуации (*контексте*) техногенной катастрофы:

- 1) запуск сервера агентов и портала агентов;
- 2) задание запроса на естественном языке;
- 3) разбор запроса транслирующим агентом (результат в виде XML файла) (рис. 7);
- 4) представление результатов обработки в табличной форме (регулярные выражения, имена классов и атрибутов онтологии);
- 5) представление результатов обработки в табличной форме (таблицы в развернутом виде) (рис. 8);
- 6) пример взаимодействия агентов;

- 7) передача результатов разбора запроса транслирующим агентом агенту управления онтологиями;
- 8) сравнение словаря запроса и словаря онтологии (рис. 9);
- 9) создание абстрактного контекста (рис. 10);
- 10) взаимодействие агента управления контекстом с интерфейсными агентами;
- 11) прикладной контекст (рис. 11).

5. Заключение

На основании анализа существующих проектов в области управления контекстом определены свойства контекста и разработана спецификация современных требований для управления контекстом, положенная в основу технологии управления контекстом. Разработанная технология ориентирована на интеллектуальную поддержку принятия решений в открытой информационной среде и на использование однородного представления (онтологии) для описания контекстов.

В качестве формальной модели представления контекста используется модель объектно-ориентированных сетей ограничений. Используемый формализм описания контекста совместим со стандартизируемым языком OWL, для описания онтологий, и удобно представляется средствами языка XML. Методология построения технологии основана на создании абстрактного контекста и его последующей реализации в виде прикладного контекста для заданных требований пользователя, ограничений со стороны источников знаний и условий задачи. Методология заключается в динамической интеграции контекстов взаимодействующих при принятии решения объектов в контекст задачи с дальнейшим решением данной задачи как задачи удовлетворения ограничений. Для апробации разработанной методологии создан исследовательский прототип многоагентной среды управления контекстом для интеллектуальной поддержки принятия решений, основанный на модели FIPA. Определены агенты среды, их функции, предложен и апробирован общий сценарий взаимодействия агентов для решения задачи пользователя с момента поступления запроса до генерации ответа пользователю на основе моделей абстрактного и прикладного контекстов.

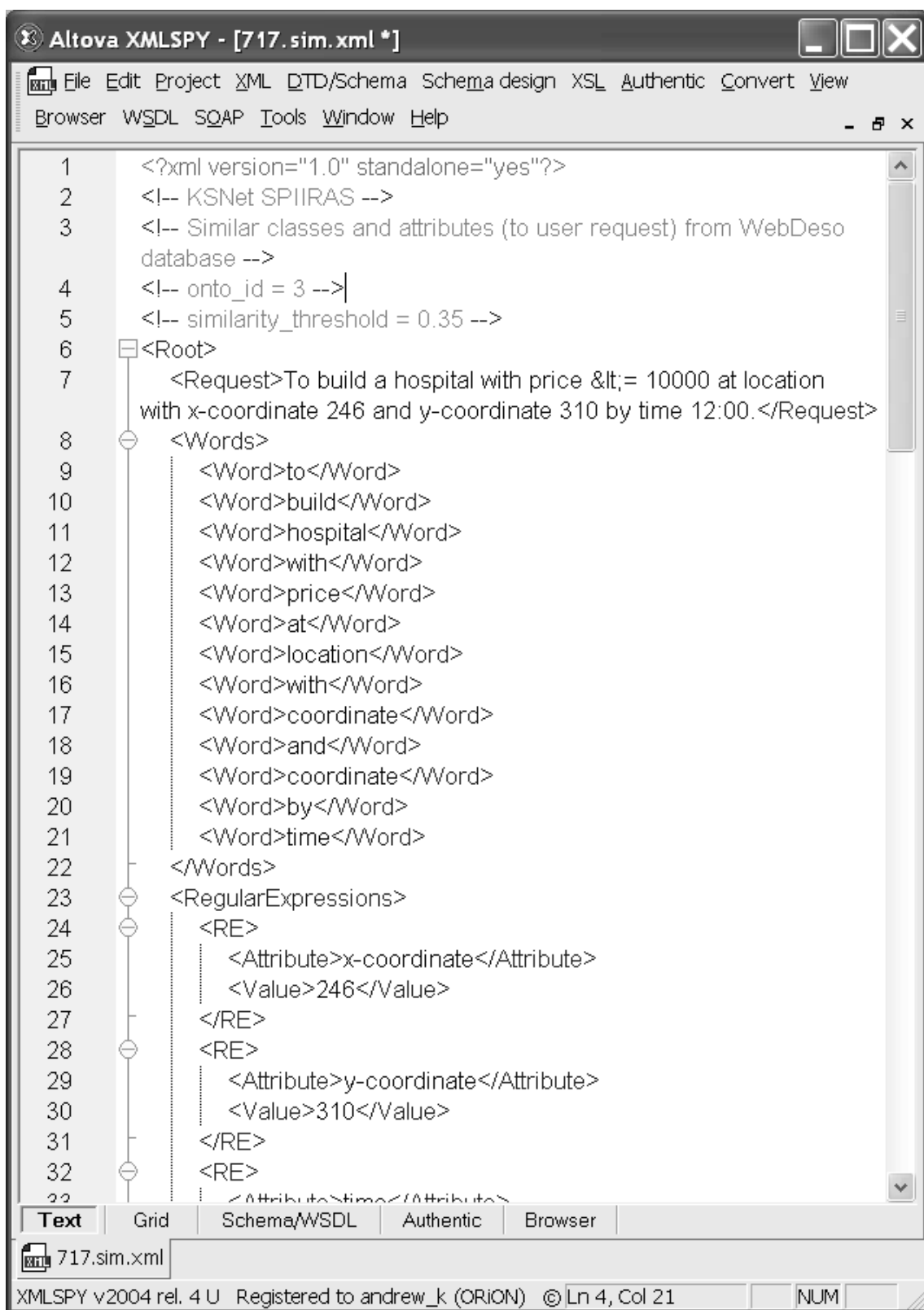


Рис. 7. Разбор запроса транслирующим агентом (результат в виде XML файла).

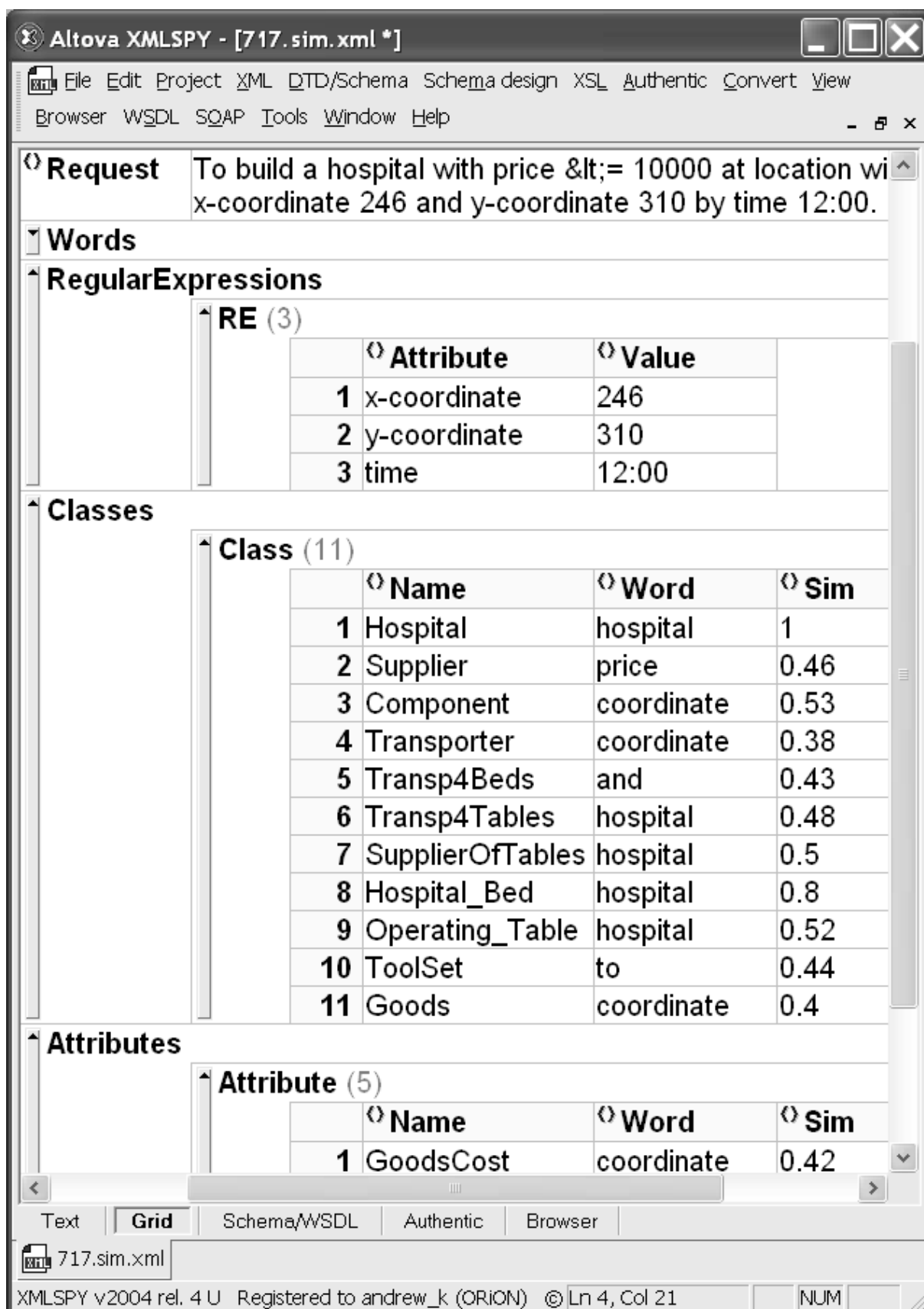


Рис. 8. Представление результатов обработки в табличной форме (таблицы в развёрнутом виде).

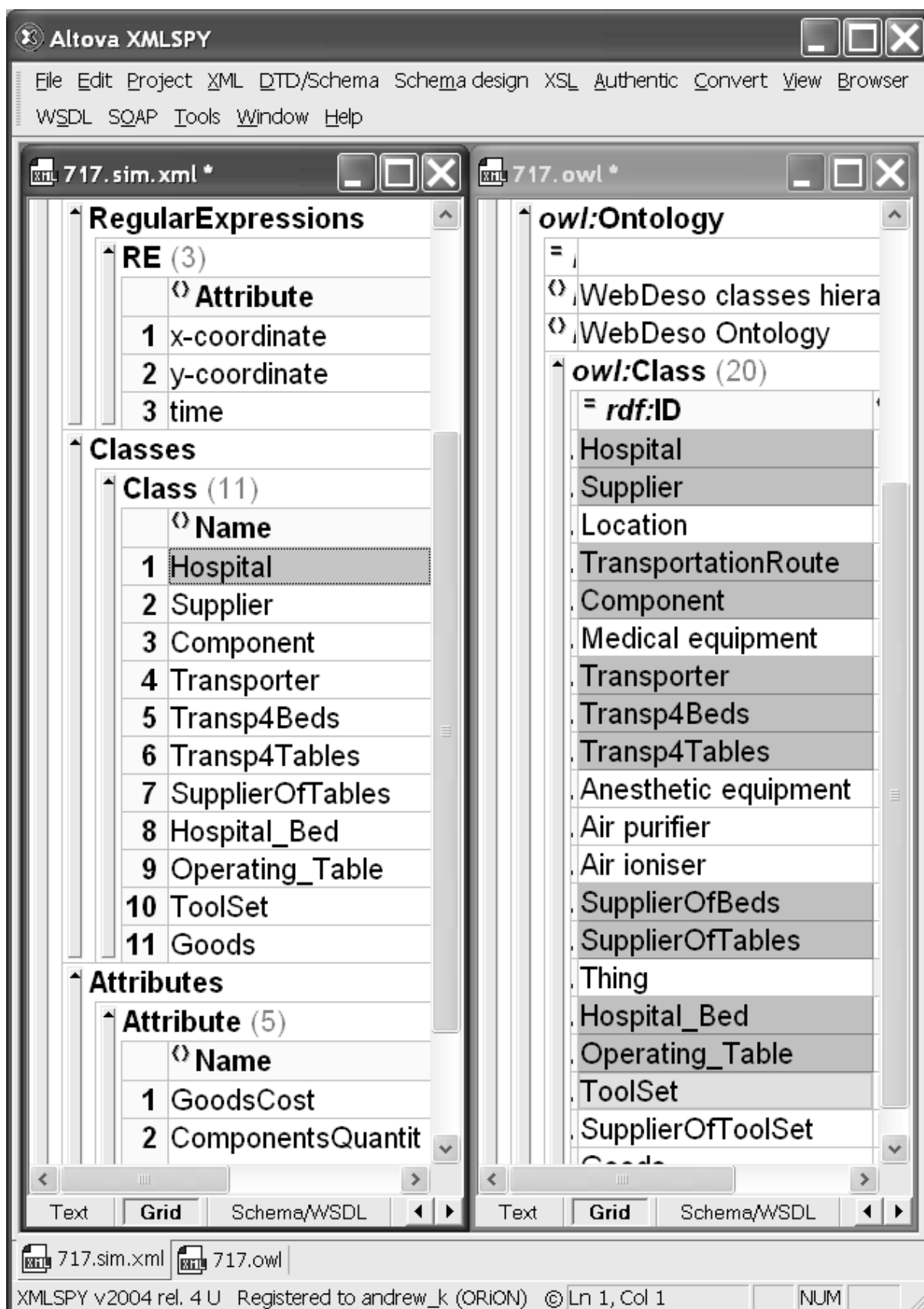


Рис. 9. Сравнение словаря запроса и словаря онтологии.

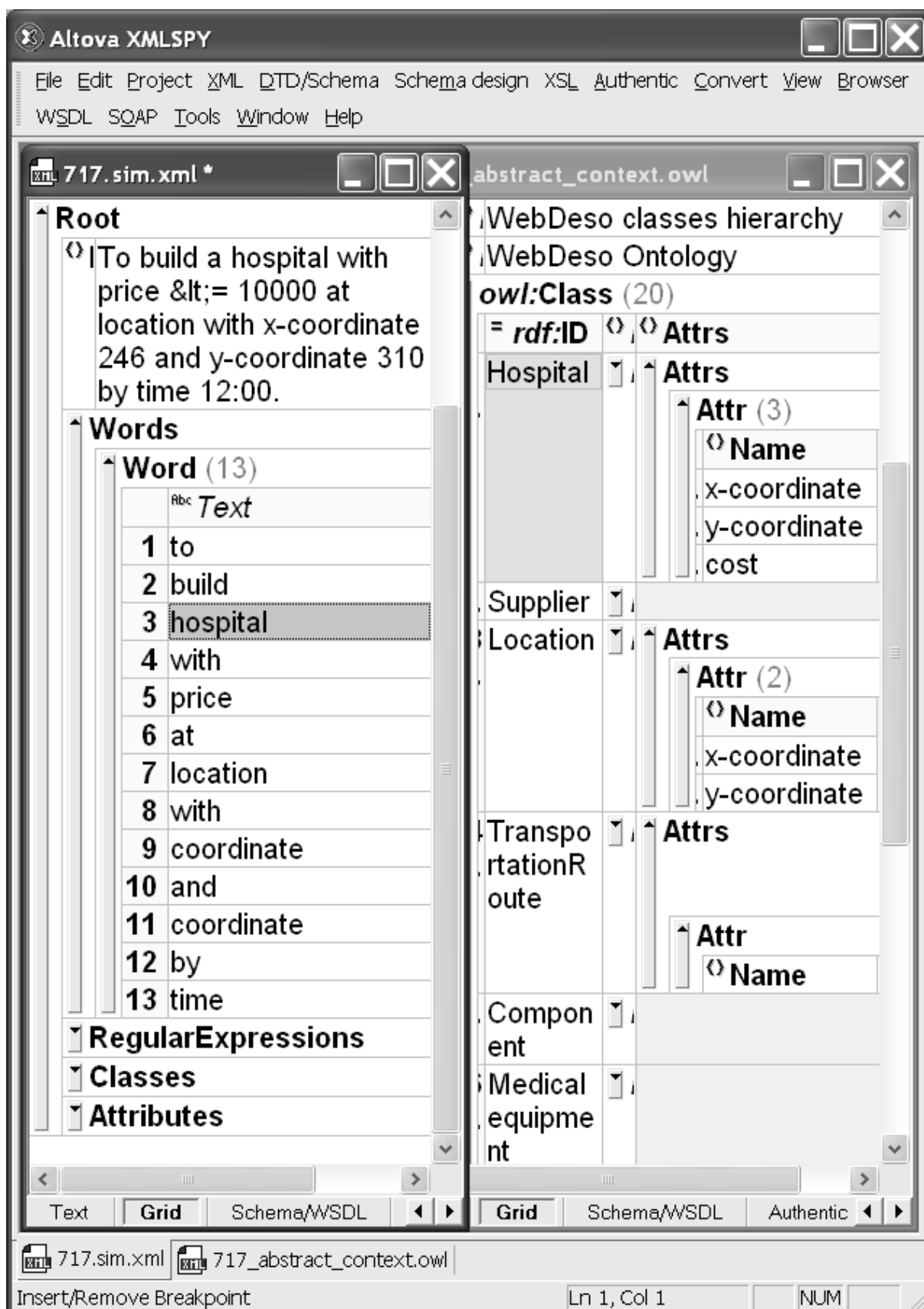


Рис. 10. Создание абстрактного контекста.

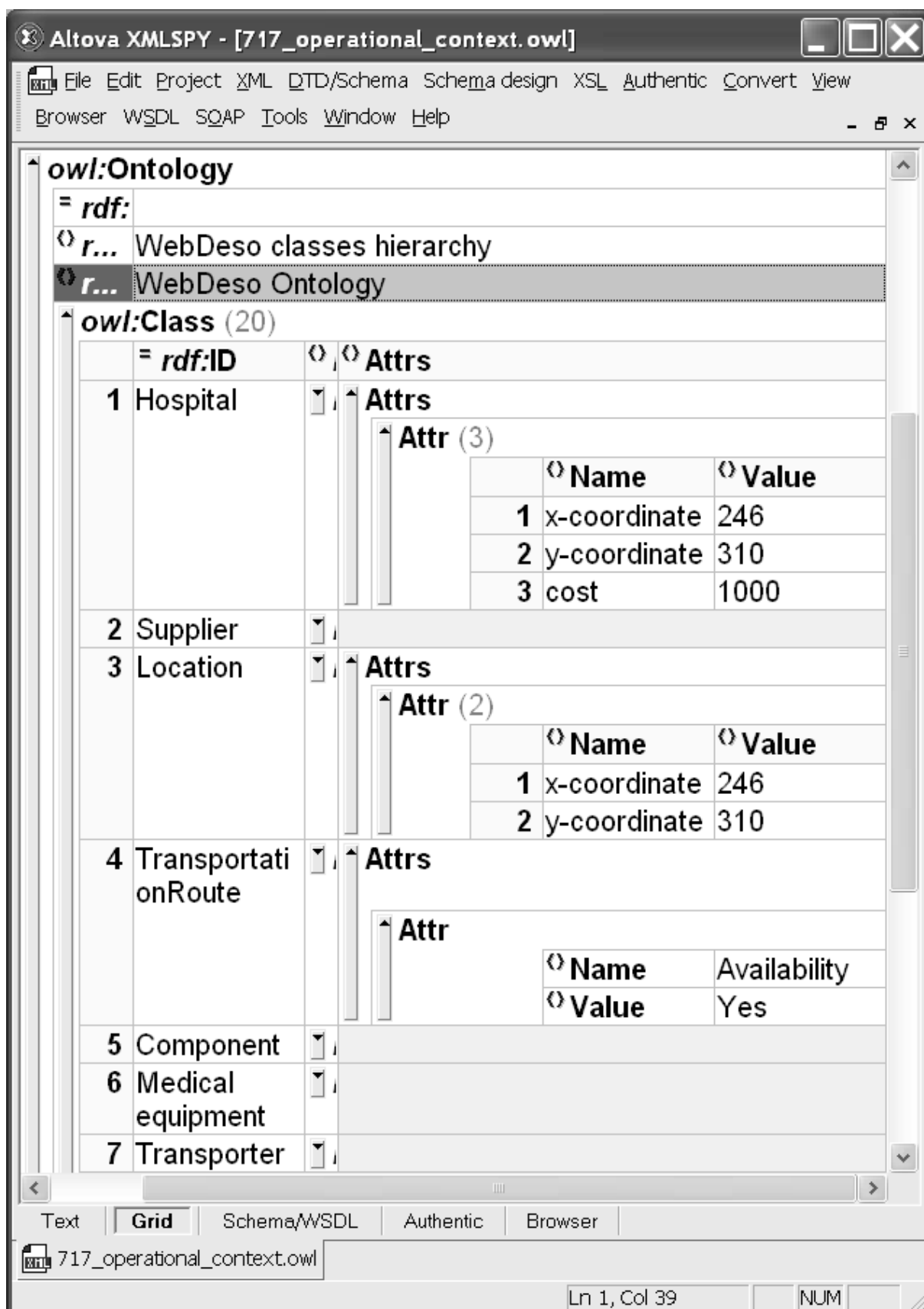


Рис. 11. Прикладной контекст.

Литература

1. *Brézillon P.* Context in Artificial Intelligence: I. A Survey of the Literature // *Computer & Artificial Intelligence*, 1999. Vol. 18, no. 4. P. 321–340.
2. *Brézillon P.* Context in Artificial Intelligence: II. Key Elements of Contexts // *Computer & Artificial Intelligence*, 1999. Vol. 18, no. 5. P. 425–446.
3. *Barrett K., Power R.* State of the Art: Context Management [Electronic resource] // M-Zones Research Programme, 2003. Deliverable 1.1. P. 69–87 // <http://www.m-zones.org/deliverables/dl_1/papers/4-01-Context.pdf> (on 05.06.2003)
4. *Gwizdka J.* What's in the context? [Electronic resource] // Proceedings of the CHI2000 Workshop on “The What, Who, Where, When, Why, and How of Context Awareness”, The Hague, April 1–6, 2000. // <<ftp://ftp.cc.gatech.edu/pub/gvu/tr/2000/00-18e.pdf>> (on 17.11.2003).
5. *Ebling M. R., Hunt G. D. H., Lei H.* Issues for Context Services for Pervasive Computing [Electronic resource] // Workshop on Middleware for Mobile Computing. Heidelberg, Germany, November 2001. – // <http://www.cs.arizona.edu/mmc/13_Ebling.pdf> (on 17.11.2003).
6. *Ranganathan A., Lei H.* Context-Aware Communication // *Computer*. 2003. No. 4. P. 90–92.
7. *Brézillon P.* Context in Problem Solving: a Survey // *The Knowledge Engineering Review*. 1999. Vol. 14, no. 1. P. 1–34.
8. *Shahar Y.* Dynamic Temporal Interpretation Contexts for Temporal Abstraction // *Annals of Mathematics and Artificial Intelligence*. 1998. No. 22. P. 159–192.
9. *Rakotonirainy A., Loke S., Fitzpatrick G.* Context-awareness for the Mobile Environment: Position Paper for CHI2000 Workshop on “The What, Who, Where, When, Why, and How of Context Awareness”, The Hague, April 1–6, 2000 // <<ftp://ftp.cc.gatech.edu/pub/gvu/tr/2000/00-18j.pdf>> (on 17.11.2003).
10. *McCarthy J.* Notes on Formalizing Context // Proceedings of the International Joint Conference on Artificial Intelligence. Chambéry, France, August 28–September 3, 1993. Morgan Kaufmann, 1993. Vol. 1. P. 555–560.
11. *Pomerol J.-Ch., Brézillon P.* About some relationships between knowledge and context // Modeling and Using Context (CONTEXT-01). Lecture Notes in Computer Science, Springer Verlag, 2001. P. 461–464.
12. *Brézillon P.* Context-based Modelling of Procedures and Practices in Business Environments [Electronic resource] // Report of the 2 COCONET Workshop “Roadmap Elaboration and Construction”. Finland, Helsinki-Espoo, 2–3 December, 2002. // <https://doc.telin.nl/dscgi/ds.py/Get/File-28084/COCONET_D3.2_Report_of_Workshop_2.pdf> (on 17.11.2003).
13. *Pomerol J.-Ch., Brézillon P.* Context proceduralization in decision making // Modeling and Using Context (CONTEXT-03) / Eds. by P. Blackburn, C. Ghidini, R. M. Turner, F. Giunchiglia. Lecture Notes in Artificial Intelligence, Springer Verlag, 2003. Vol. 2680. P. 491–498.
14. *Ozturk P., Aamodt A.* A context model for knowledge-intensive case-based reasoning // International Journal on Human-Computer Studies. Special Issue on Using Context in Applications. 1998. Vol. 48, no. 3. P. 331–355.
15. CYC Upper Ontology Web-site, 1997 // <<http://www.cyc.com/cyc-2-1/toc.html>>.
16. *Brézillon P.* Context dynamic and explanation in contextual graphs // Modeling and Using Context (CONTEXT-03) / Eds. by P. Blackburn, C. Ghidini, R.M. Turner, F. Giunchiglia. Lecture Notes in Artificial Intelligence, Springer Verlag, 2003. Vol. 2680. P. 94–106.
17. *Walther E., Eriksson H., Musen M. A.* Plug-and-Play: Construction of Task-Specific Expert-System Shells Using Sharable Context Ontologies // Proceedings of the AAAI Workshop on Knowledge Representation Aspects of Knowledge Acquisition. USA, San Jose, CA, 1992. AAAI, 1992. P. 191–198.
18. *Dey A. K., Salber D., Abowd G. D.* A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications [Electronic resource] // Context-Aware Computing. A Special Triple Issue of Human-Computer Interaction / Eds. by T.P. Moran, P. Dourish. Lawrence-Erlbaum, 2001. Vol. 16. // <<http://www.cc.gatech.edu/fce/ctk/pubs/HCIJ16.pdf>> (on 22.02.2004).
19. *Левашова Т. В., Пашкин М. П., Смирнов А. В., Шилов Н. Г.* Управление онтологиями // Известия РАН. Сер. Теория и системы управления. 2003. Часть 1, № 4. С. 132-146;

- часть 2, № 5. С. 89-101.
20. *Decker S., Erdmann M., Fensel D., Studer R.* Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information // *Semantic Issues in Multimedia Systems / Eds. by R. Meersman et al. Proceedings of DS-8. Kluwer Academic Publisher, Boston, 1999. P. 351–369.*
 21. *Baclawski K., Matheus C., Kokar M., Letkowski J., Kogut P.* Towards a Symptom Ontology for Semantic Web Applications // *Proceedings of Third International Semantic Web Conference. – Hiroshima, Japan, 2004. P. 650–667.*
 22. The Semantic Web Community Portal, 2003 // <<http://www.semanticweb.org/>>.
 23. W3C: World Wide Web Consortium, 2003 // <<http://www.w3.org/>>.
 24. FIPA Agent Management Specification. Foundation for Intelligent Physical Agents (FIPA) – № SC00023J, 2002. // <<http://www.fipa.org/>>.
 25. FIPA Ontology Service Specification. Part 12 – Foundation for Intelligent Physical Agents (FIPA). № OC00006. Version 1.0, 1998. // <<http://www.fipa.org/>>.
 26. Extensible Markup Language (XML). W3C Architecture Domain, W3C, 2002 // <<http://www.w3.org/XML/>>.
 27. *Miller G. A., Beckwith R., Fellbaum C., Gross D., and Miller K. J.* Introduction to WordNet: an on-line lexical database [Electronic resource] // *International Journal of Lexicography. 1990. Vol. 4, no. 3. P. 235–244.*
 28. *Miller A. G.* Wordnet: A lexical database for English [Electronic resource] // *Communications of the ACM, 1995. Vol. 38, no. 11. P. 39–41.*
 29. Wordnet, 2002. URL: <<http://www.cogsci.princeton.edu/~wn/>>.
 30. *Tomuro N.* Semi-automatic Induction of Systematic Polysemy from WordNet // *Proceedings of the Workshop on Usage of WordNet in Natural Language Processing Systems of the 17th International Conference on Computational Linguistics (COLING-98), 1998. Montreal, Canada. P. 108–114.*
 31. *Lytinen S., Tomuro N., Repede T.* The Use of WordNet Sense Tagging in FAQFinder // *Proceedings of the Workshop on Artificial Intelligence for Web Search of the 17th National Conference on Artificial Intelligence (AAAI-2000), 2000. Austin, USA.*
 32. *Gangemi A., Guarino N., Masolo C., Oltramari A.* Restructuring WordNet's Top-Level: The OntoClean approach // *Proceedings of the Workshop on Ontologies and Lexical Knowledge Bases (OntoLex 2002) of the third International Conference on Language Resources and Evaluation (LREC 2002), 2002. Las Palmas, Spain.*
 33. WonderWeb – Ontology Infrastructure for the Semantic Web, 2002 // <<http://wonderweb.semanticweb.org/>>.
 34. ActivePerl. The Industry-Standard Perl Distribution for Linux, Solaris, and Windows. ActiveState, 2002. // <<http://www.activestate.com/Products/ActivePerl/>>.
 35. ActiveState Homepage. ActiveState, 2002 // <<http://www.activestate.com/>>.
 36. *Городецкий В. И., Карсаев О. В., Котенко И. В., Хабалов А. В.* MAS DK: инструментарий для разработки многоагентных систем и примеры приложений // *Труды Международного конгресса «Искусственный интеллект в XXI веке». Россия, Дивноморское, 2001. С. 249–262.*
 37. *Смирнов А. В., Пашкин М. П., Шилов Н. Г., Левашова Т. В., Крижановский А. А.* Реализация проблемно-ориентированных агентов логистики знаний // *Труды Международных научно-технических конференций «Интеллектуальные системы», «Интеллектуальные САПР» (IEEE AIS'03, CAD-2003). Россия, Дивноморское, 3–10 сентября, 2003. М.: Физматлит, 2003. Т. 1. С. 464–469.*
 38. *Smirnov A., Pashkin M., Chilov N., Levashova T.* KNet-Approach to Knowledge Fusion from Distributed Sources // *Computing and Informatics, 2003. Vol. 22. P. 105–142.*
 39. *Смирнов А. В., Пашкин М. П., Шилов Н. Г., Левашова Т. В., Крижановский А. А.* Принципы реализации проблемно-ориентированных агентов в системе интеграции знаний // *Проблемы управления и моделирования в сложных системах. Труды V Международной конференции. Самара, Россия, 17–21 июня, 2003. С. 284–294.*
 40. *Смирнов А. В., Пашкин М. П., Шилов Н. Г., Левашова Т. В., Крижановский А. А.* Методология управления контекстом при формировании контекста задачи для интеллектуальной поддержки принятия решений // *Труды Международных научно-технических конференций «Интеллектуальные системы», «Интеллектуальные САПР», 2004. Дивноморское, Россия, 3–10 сентября. М.: Физматлит, 2004. Т. 1. С. 321–326.*
 41. *Смирнов А. В., Пашкин М. П., Шилов Н. Г., Левашова Т. В., Крижановский А. А.* Формиро-

- вание контекста задачи для интеллектуальной поддержки принятия решений. Фундаментальные основы информационных технологий и систем // Труды Института системного анализа РАН. 2004. М.: ИСА РАН, 2004. Т. 9. С. 125–188.
42. *Смирнов А. В., Пашкин М. П., Шилов Н. Г., Левашова Т. В.* Использование контекста при Формировании модели задачи для интеллектуальной поддержки принятия оперативных решений // Материалы докладов V-й Международной конференции «Интеллектуальные системы», Россия, Дивноморское, 2005. М.: Физматлит, 2005. Т. 1. С. 356–361.
 43. *Смирнов А. В., Пашкин М. П., Шилов Н. Г., Левашова Т. В.* Методология использования контекста при интеллектуальной поддержке принятия решений // Материалы докладов 1-й Международной конференции «Системный анализ и информационные технологии», Россия, Переславль-Залесский, 12–16 сентября, 2005. С. 276–282.
 44. *Smirnov A., Pashkin M., Chilov N., Levashova T.* Agents-based Knowledge Logistics // Design of Intelligent Multi-Agent Systems / Eds. by R. Khosla, N. Ichalkaranje, L.C. Jain. New York: Springer, 2005. P. 63–101.
 45. *Smirnov A., Pashkin M., Chilov N., Levashova T., Krizhanovsky A.* Multi-Agent Decision Support System for Disaster Response and Evacuation // Monitoring, Security, and Rescue Techniques in Multiagent Systems / Eds. by B. Dunin-Keplicz, A. Jankowski, A. Skowron, M. Szczuka. Advances in Soft Computing. New York: Springer, 2005. P. 385–395.
 46. *Smirnov A., Pashkin M., Chilov N., Levashova T., Krizhanovsky A.* Agent-Based Intelligent Support to Coalition Operations: A Case Study of Health Service Logistics Support // Information & Security. An International Journal. IT in Coalition and Emergency Operations / Eds. by V. Shalamanov, G. Johnson, J. Fay. ProCon Ltd., Sofia, 2005. Vol. 16. P. 41–61.
 47. *Smirnov A., Pashkin M., Chilov N., Levashova T.* Operational Decision Support: Context-Based Approach and Technological Framework // Modeling and Using Context / Eds. by A. Dey, B. Kokinov, D. Leake, R. Turner. Proceedings of the 5th International Conference CONTEXT-05, Paris, France, July, 2005. LNAI 3554. Berlin, Heidelberg: Springer, 2005. P. 476–489.