

14. Ермолев Ю.М. Методы стохастического программирования. — М.: Наука, 1976. — 340 с.
15. Растринин Л.А. Статистические методы поиска. Теоретические основы технической кибернетики. — М.: Наука, 1968. — 378 с.
16. Robbins H., Monro S.A stochastic approximation method // Ann. Math. Stat. — 1951. — V. 22. — P. 400–407.
17. Ширяев А.Н. Элементарная теория вероятностей. Математические основания. Предельные теоремы. — М.: МЦНМО, 2004. — 519 с.
18. Ковалев М.М. Дискретная оптимизация. (Целочисленное программирование). — Минск: Изд-во БГУ, 1977. — 191 с.
19. Ху Т. Целочисленное программирование и потоки в сетях. — М.: Мир, 1974. — 520 с.
20. Кофман А., Анри-Лабродер А. Методы и модели исследования операции: Целочисленное программирование. — М.: Мир, 1977. — 432 с.

Поступила 18.04.2012 г.

УДК 004.89

МЕТОД ПРИНЯТИЯ РЕШЕНИЙ НА ОСНОВЕ АНАЛИЗА СИТУАЦИЙ И СЕМАНТИЧЕСКИХ ТЕХНОЛОГИЙ

В.В. Разин, А.Ф. Тузовский

Томский политехнический университет
E-mail: moonbreeze@sibmail.com

Предлагается подход к построению систем автоматизированного анализа ситуаций, основанный на стеке технологий Semantic Web. Описание подхода включает формальную модель описания ситуаций, описание базовых онтологий, реализующих её, и описание архитектуры системы анализа ситуаций. Приводится пример описания ситуации при помощи интерпретации предлагаемой формальной модели в терминах OWL-онтологий.

Ключевые слова:

Анализ ситуаций, Semantic Web, онтологии, OWL, продукционные правила.

Key words:

Situational analysis, Semantic Web, ontologies, OWL, productional rules.

Анализ ситуаций является одной из задач принятия решений. Автоматизация анализа ситуаций позволяет выявлять текущие состояния системы, которые могут быть важными для конечного пользователя, и требуют оперативного принятия решений. Для выполнения такого анализа требуется определить язык описания ситуаций и сформировать алгоритм выявления интересующих пользователей ситуаций. Одним из подходов к принятию решений на основе формального анализа ситуаций является метод ситуационного управления [1]. В данном методе предлагается использовать логические модели для описания ситуаций и работы с ними. В настоящее время в рамках концепции Semantic Web активно развиваются семантические технологии, использующие специальный вид семантических моделей — онтологии.

В рамках данной работы описывается подход к анализу ситуаций, основанный на применении семантических технологий.

Основной идеей предлагаемого подхода является разработка онтологий для описания ситуаций и использование продукционных правил для их анализа.

Онтология — модель предметной области, состоящая из множества понятий, множества экземпляров понятий и множества отношений (свойств). Отношения могут быть как объектными (связывающими понятия и их экземпляры друг с другом),

так и литеральными (связывающими понятия и их экземпляры с текстовыми строками). Множество понятий и отношений между ними определяют общую схему хранения данных, представленных как множество утверждений об экземплярах понятий, или аксиом онтологии. Эти утверждения, или триплеты, имеют вид «субъект—предикат—объект». Основными языками, при помощи которых описываются онтологии, являются стандартизованные консорциумом W3C языки RDF [2] и OWL [3].

Один из возможных подходов к описанию ситуаций при помощи онтологий был описан в работе [4]. В данном подходе описание ситуаций предлагается разделять на две части. Первая часть — базовая онтология описания ситуаций, представляющая абстрактную ситуацию с учётом контекста пространства-времени. Вторая часть — онтология, описывающая конкретную предметную область и относящиеся к ней возможные классы ситуаций, к которым в тот или иной момент времени может принадлежать ситуация на объекте управления. Онтология предметной области в этом случае будет расширением (надстройкой) над базовой онтологией и будет содержать понятия, наследующие понятиям базовой онтологии. Исходные данные о ситуации, поступающие в систему, например, с датчиков, при этом преобразуются в экземпляры понятий онтологии предметной области и отношения между ними.

Для определения закономерностей взаимодействия элементов управляемой системы, а также описания изменений их состояния с течением времени используются продукционные правила вида «если ... то ...». Набор правил, задаваемых пользователем, загружается в систему логического вывода, которая на основе содержащихся в онтологии утверждений создаёт согласно этим правилам новые экземпляры отношений более высокого уровня, соответствующие искомым ситуациям.

Продукционные правила можно задавать как в виде правил прямого, так и в виде правил обратного логического вывода. В первом случае система логического вывода работает «от фактов к цели», применяя к содержащимся в онтологии данным все допустимые правила до тех пор, пока имеется возможность применять эти правила (т. е. пока есть данные, соответствующие части «если», используемого правила). При выполнении обратного логического вывода действия системы вывода выполняются в обратном направлении, «от цели» правила. Для каждого из элементов блока «если» в онтологии ищутся соответствующие ему утверждения либо, если существуют правила, в которых этот элемент стоит в блоке «то», обратный вывод рекурсивно применяется к этим правилам. При задании системы продукционных правил можно комбинировать оба этих подхода. Для описания наборов правил можно использовать существующие стандартизованные языки описания продукционных правил для машин вывода на онтологиях, например, такие как SWRL [5] или RIF [6].

В рамках предлагаемого подхода была учтена необходимость представлять описание структуры объекта управления в контексте пространства–времени, а также возможная неполнота знаний об элементах этого описания. В связи с этим под ситуацией S в данном случае понимается совокупность всех сведений о текущем и всех предыдущих состояниях объекта управления, или система $\langle O, R, E, F, Rul \rangle$, где O – множество физических объектов, входящих в состав управляемой системы, R – множество областей в пространстве, E – множество событий, происходящих в управляемой системе, F – множество предикатов, отражающих связи между элементами множеств O , R , и E , а Rul – множество продукционных правил вида $(F \Rightarrow G)$ и $(F \Leftrightarrow G)$, описывающих базовые механизмы логического вывода, в том числе ограничения на значения предикатов F , а также предусловия и следствия для происхождения событий E .

Фактически, в такой интерпретации ситуацию S можно обозначить как полное состояние объекта управления. При этом под классом ситуаций C подразумевается некоторый кортеж $\langle O', R', E', F' \rangle$, где O' , R' , E' – это множества переменных, обозначающих элементы множеств O , R , E соответственно, а F' – подмножество F , каждый элемент которого – предикат, аргументами которого являются элементы множеств O' , R' и E' . Класс ситуаций фактически является шаблоном, описывающим отноше-

ния между частью элементов объекта управления. Таким образом, ситуация на объекте управления анализируется с целью выявления тех классов ситуаций, к которым она принадлежит. Обозначим как обоснование класса ситуаций C_B интерпретацию класса ситуаций C на множествах $\langle O, R, E, F \rangle$.

Иначе говоря, обоснованием класса ситуаций C будет такая подстановка элементов O , R и E вместо переменных из O' , R' и E' , при которой все флюенты из F' будут истинны.

Будем говорить, что ситуация S принадлежит к классу C , если S содержит обоснование C . Иначе говоря, если C – шаблон, то C_B – часть описания ситуации, подходящая к этому шаблону.

Как было упомянуто выше, описание ситуации включает в себя множество областей в пространстве R . С помощью предикатов F эти области могут быть ассоциированы с объектами и событиями (элементами множеств O и E соответственно). Таким образом, знания о том, где находятся объекты и где происходят события, содержатся, в первую очередь, в описании областей в пространстве (элементов R), соответствующих объектам и событиям, а взаимное расположение объектов и областей, в которых действуют те или иные события, может быть логически выведено. Качественные отношения взаимного расположения областей в пространстве основываются, с одной стороны, на исчислении соединения областей (RCC – region connection calculus) [7] и, с другой стороны, на псевдо-физических логиках, описанных в работе [1].

На рис. 1 и рис. 2 соответственно представлены схемы общей онтологии описания ситуации и диаграмма импортов онтологий, формирующих базовую онтологию описания ситуаций.

Непременным атрибутом события является время его наступления, а также задействованные в нём объекты. Это выражается предикатом $holds(e_1, o_1, o_2, \dots, o_n, \tau_1)$, где τ_1 соответствует промежутку времени, в течение которого происходит событие. Также событие можно ограничить некоторой областью в пространстве: $holds(e_1, o_1, o_2, \dots, o_n, r_1, \tau_1)$. Предусловия и следствия выражаются при помощи продукционных правил – элементов множества правил Rul . Возможность наступления события в контексте пространства–времени выражается предикатом $possible: possible(e_1, o_1, o_2, \dots, o_n, \tau_1)$.

Пример правила для предусловия:

$$f_1(o_1, o_2, \tau_1) \wedge f_2(o_1, o_3, \tau_2) \wedge f_3(o_4, o_5, \tau_3) \Rightarrow \\ \Rightarrow possible(e_1, o_1, o_2, o_3, o_4, o_5, \tau_1).$$

Пример правила для следствия:

$$holds(e_1, o_1, o_2, o_3, o_4, o_5, \tau_1) \Rightarrow \\ \Rightarrow f_1(o_1, o_2, \tau_1) \wedge f_2(o_1, o_3, \tau_2) \wedge f_3(o_4, o_5, \tau_3).$$

В рамках данного подхода события можно разделить на три группы: внутренние (события, происходящие внутри объекта управления), внешние (события, наступление которых вызвано извне объекта управления), и события-действия (собы-

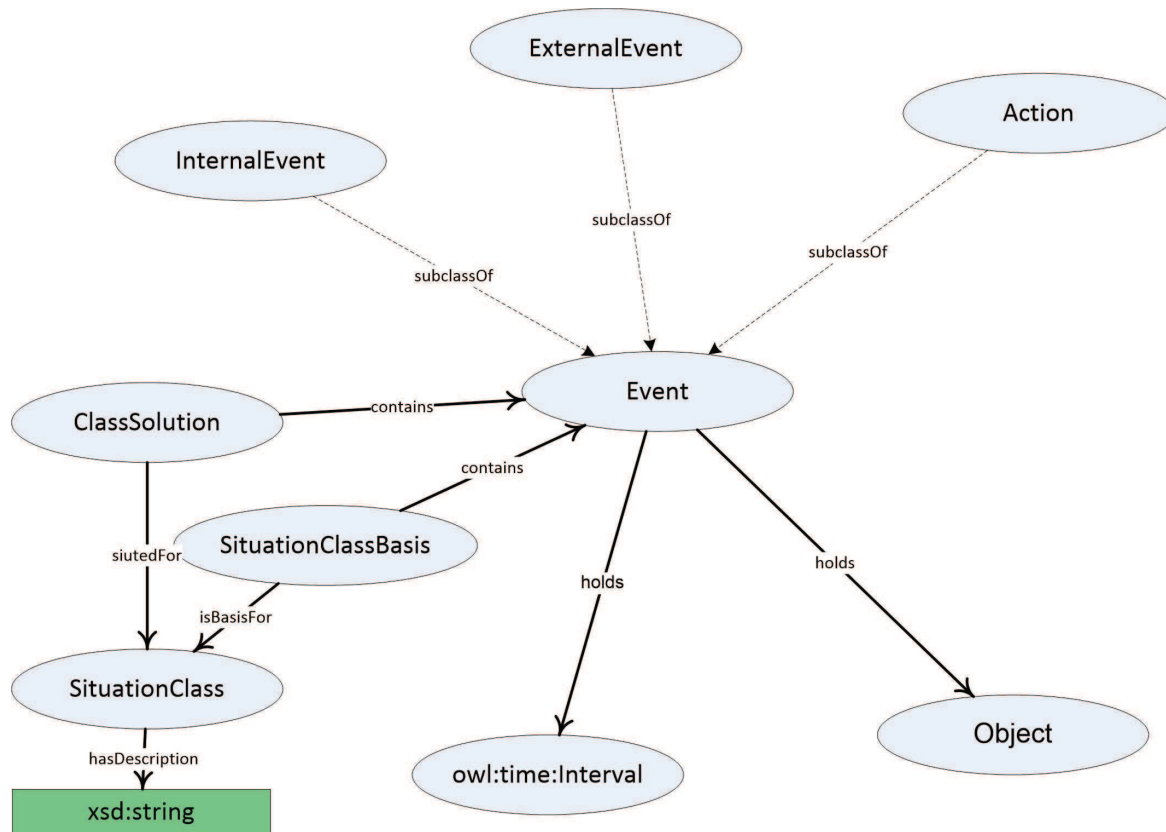


Рис. 1. Общая онтология описания ситуаций

тия, наступление которых вызвано человеком). Каждый вид событий обладает своей спецификой: внутренние события описывают поведение элементов управляемой системы и, следовательно, обладают как предусловием, так и следствием, внешние события исходят извне системы и, не являясь предсказуемыми, обладают только следствием, а события-действия обладают и предусловием, и следствием, однако предполагают некоторую свободу воли и потому не обязательно наступают немедленно по выполнении своего предусловия.

Будем говорить, что событие e_i переводит ситуацию S в/из класса C , если в результате его наступления ситуация S стала/перестала принадлежать этому классу (или если в результате его наступления S стала/перестала включать обоснование C).

Фактически, в рамках данного подхода события инкапсулируют в себе всю динамику объекта управления: любое изменение состава его элементов или отношений между ними (то есть изменения множеств O , R , E , F) соответствует некоторому событию, и наоборот.

Обозначим как решение для класса ситуаций C совокупность E_{sol} событий, при наступлении которых в некоторой ситуации S , принадлежащей классу ситуаций C , данная ситуация перестаёт принадлежать этому классу. При этом решения могут быть в данный момент времени доступными и недоступны-

ми. Доступным решением является такая совокупность событий $E_{sol}=\{e_1, e_2, e_3, \dots, e_n\}$, которые для каждого e_i являются истинными: $holds(e_i, o_1, o_2, \dots, o_n, \tau_N)$ (напомним, что τ_N – переменная, соответствующая данному моменту времени) или $possible(e_i, o_1, o_2, \dots, o_n, \tau_N)$ для событий-действий.

Общая онтология описания ситуаций (рис. 1) включает понятия события $Event$ с дочерними понятиями $InternalEvent$ (внешнее событие), $ExternalEvent$ (внутреннее событие) и $Action$ (действие), понятие $Object$, а также понятия $SituationClass$ (класс ситуаций), $SituationClassBasis$ (обоснование класса ситуаций) и $ClassSolution$ (решение для класса ситуаций). Время происхождения события $Event$ описывается при помощи понятия $Interval$ онтологии $OWL-Time$ [8].

Онтология RCC (рис. 2) определяет отношения исчисления соединения областей, $Spatial$ и $QualitativeSpatial$ – области в пространстве и качественные отношения между ними соответственно, $owl:time$ – онтология описания времени [8], $Interval$ – онтология, описывающие отношения интервальной алгебры Аллена. Онтология $Core$ сводит все элементы описания ситуации воедино и, по существу, является базовой онтологией описания ситуаций, которой будет наследовать любая онтология, описывающая конкретную предметную область, классы ситуаций и решения в ней.

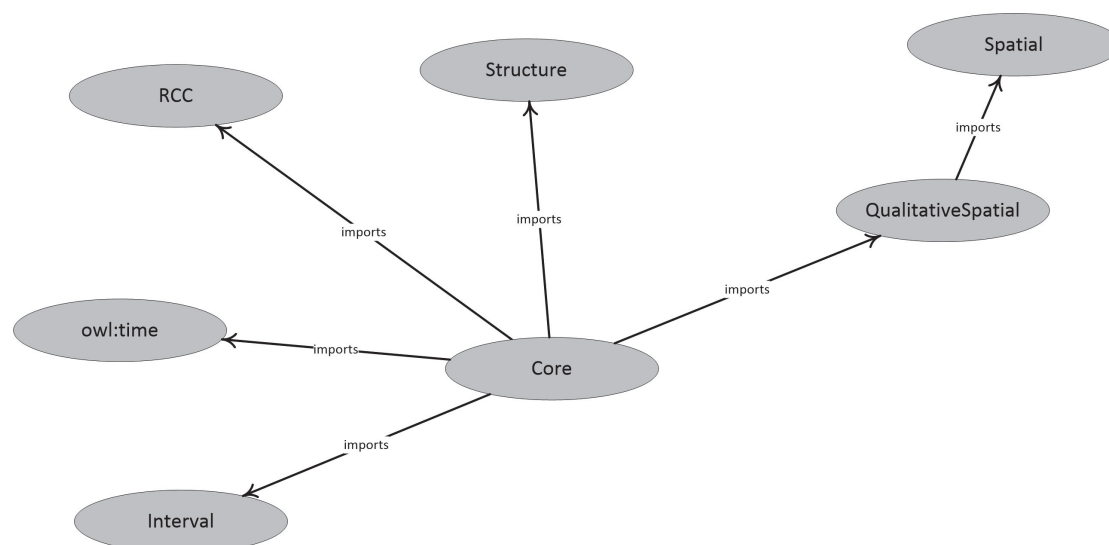


Рис. 2. Схема импорта онтологий

Основные элементы системы анализа ситуаций на основе описываемого подхода представлены на рис. 3.

Приведём описание тестовой ситуации. Пусть требуется описать выезд такси к клиенту. Клиент находится на указанной им точке подбора, при этом какое-либо свободное такси должно доехать туда возможно быстрее и подобрать пассажира (таксист подъезжает к точке и либо сразу подбирает пассажира, либо сообщает о прибытии диспетчеру, который звонит клиенту).

Допустим, что клиент будет ждать 15 минут, а потом отменит заказ и обратится в другое такси. Введём в онтологию описания ситуации понятия Client, Taxi, и PickupPoint (точка подбора). Обобщим недопустимую в данном примере ситуацию до класса ситуаций «ожидание клиентом такси в течение 15 минут», обозначив его как ClientWaiting1. Ниже приведены правила, описывающие такой класс ситуаций, согласно которым система его распознает (здесь и далее продукционные правила приводятся в нотации SWRL).

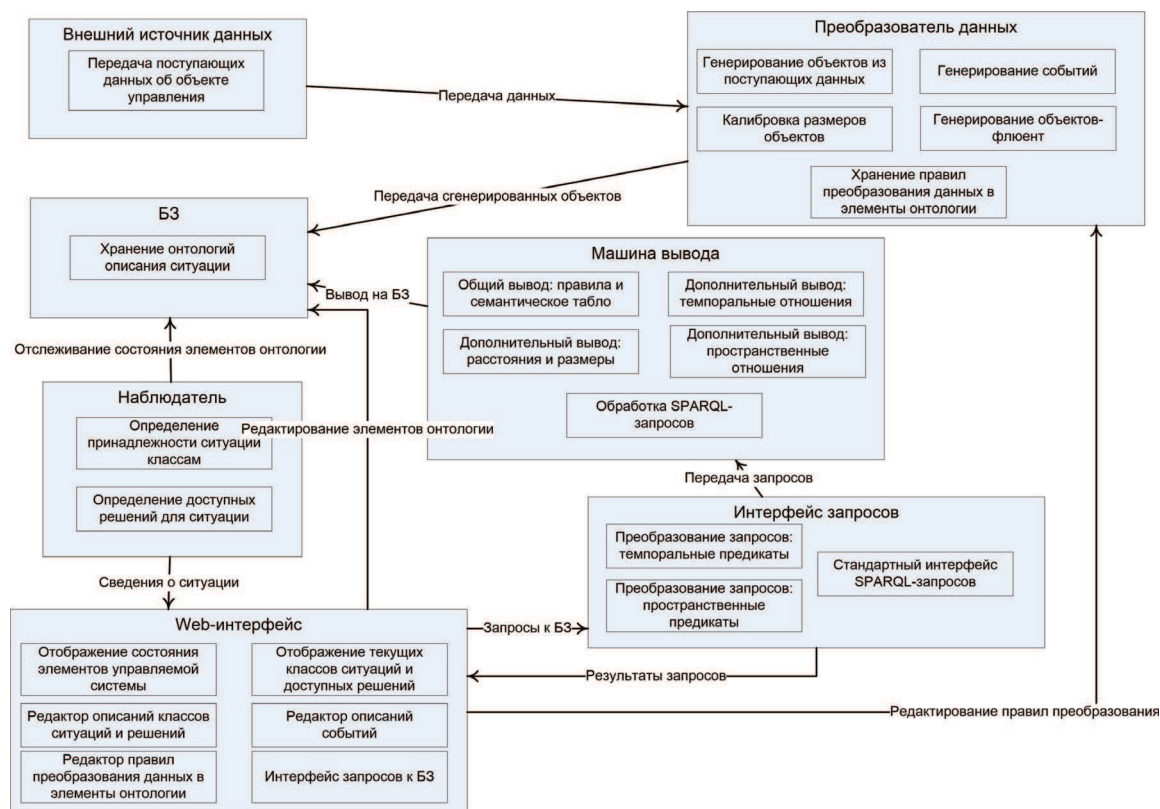


Рис. 3. Общая схема системы анализа ситуаций

$\text{ClientWaiting1}(\text{?sitclass}) \wedge \text{Client}(\text{?cl1}) \wedge$
 $\text{PickupPoint}(\text{?PP}) \wedge \text{Interval}(\text{?tp1}) \wedge \text{WaitEvent}(\text{?we1}) \wedge$
 $\text{holds}(\text{?we1}, \text{cl}) \wedge \text{holds}(\text{?we1}, \text{?PP}) \wedge$
 $\text{holds}(\text{?we1}, \text{?tp1}) \wedge \text{duration}(\text{?tp1}, \text{?dur}) \wedge$
 $\text{greaterThan}(\text{?dur}, 15) \wedge \text{hasBeginning}(\text{?tp1}, \text{?timepoint1}) \wedge$
 $\text{subtract}(\text{?res}, \text{Now}, \text{?timepoint1}) \wedge$
 $\text{lessThan}(\text{?res}, 15) \Rightarrow \text{active}(\text{?sitclass1}, \text{true}) \wedge$
 $\text{makeInstance}(\text{?CWBasis}) \wedge$
 $\text{ClientWaitingBasis}(\text{?CWBasis}) \text{ contains}(\text{?CWBasis}, \text{cw1}).$

Здесь WaitEvent – событие, заранее определённое как:

$\text{Client}(\text{?cl}) \wedge \text{Location}(\text{?loc}) \wedge \text{Interval}(\text{?tp1}) \wedge$
 $\wedge \text{waiting}(\text{?cl}, \text{true}) \wedge \text{waitingTime}(\text{?cl}, \text{?tp1}) \wedge$
 $\wedge \text{occupies}(\text{?cl}, \text{?loc}) \Rightarrow \text{makeInstance}(\text{?waitEvent}) \wedge$
 $\wedge \text{WaitEvent}(\text{?waitEvent}) \wedge$
 $\wedge \text{holds}(\text{?waitEvent}, \text{?cl}) \wedge \text{holds}(\text{?waitEvent}, \text{?loc}) \wedge$
 $\wedge \text{holds}(\text{?waitEvent}, \text{?tp1}).$

Опишем возможное решение для класса ситуаций ClientWaiting1. Наиболее очевидным решением будет подбор клиента на точке с ожиданием прибытия клиента в течение, к примеру, 10 минут. Создадим экземпляр понятия ClassSolution и понятие PickupEvent, наследующее понятию InternalEvent. Создадим также экземпляр PickupEvent и пометим его как часть решения для класса ситуаций

ClientWaiting1. Полученная онтология показана на рис. 4.

Приведём описание события PickupEvent:

$\text{Taxi}(\text{?taxi1}) \wedge \text{Client}(\text{?cl1}) \wedge \text{waiting}(\text{?cl1}) \wedge$
 $\text{ExecutePickupOrderEvent}(\text{?pickupOrder}) \wedge$
 $\wedge \text{VeryCloseSituatingEvent}(\text{?vCloseSit}) \text{ Interval}(\text{?tp1}) \wedge$
 $\wedge \text{Interval}(\text{?tp2}) \wedge \text{holds}(\text{?pickupOrder}, \text{?taxi1}) \wedge$
 $\wedge \text{holds}(\text{?pickupOrder}, \text{?cl1}) \wedge \text{holds}(\text{?pickupOrder}, \text{?tp1}) \wedge$
 $\wedge \text{holds}(\text{?vcloseSit}, \text{?taxi1}) \wedge \text{holds}(\text{?vCloseSit}, \text{?cl1}) \wedge$
 $\wedge \text{holds}(\text{?vCloseSit}, \text{?tp2}) \wedge$
 $\wedge \text{inside}(\text{?tp2}, \text{?tp1}) \Rightarrow \text{makeInstance}(\text{?pickup}) \wedge$
 $\wedge \text{makeInstance}(\text{?tp3}) \wedge \text{PickupEvent}(\text{?pickup}) \wedge$
 $\text{Interval}(\text{?tp3}) \wedge \text{holds}(\text{?pickup}, \text{?taxi1}) \wedge$
 $\wedge \text{holds}(\text{?pickup}, \text{?cl1}) \wedge \text{holds}(\text{?pickup}, \text{?tp3}) \wedge$
 $\text{hasBeginning}(\text{?tp3}, \text{?arrival}) \wedge \text{add}(\text{?result}, \text{?arrival}, 10) \wedge$
 $\wedge \text{hasEnd}(\text{?tp3}, \text{?result}).$

Упомянутые в описании события VeryCloseSituatingEvent и ExecutePickupOrderEvent указывают на то, что для выполнения события PickupEvent необходимо, чтобы такси находилось на близком расстоянии от клиента и выполняло распоряжение подобрать его, соответственно. Формально распоряжение подобрать клиента можно разделить на два: прибыть в определённую точку и, собственно, подобрать клиента. Рассмотрим первое из этих распоряжений (обозначим как событие MovementOrderEvent):

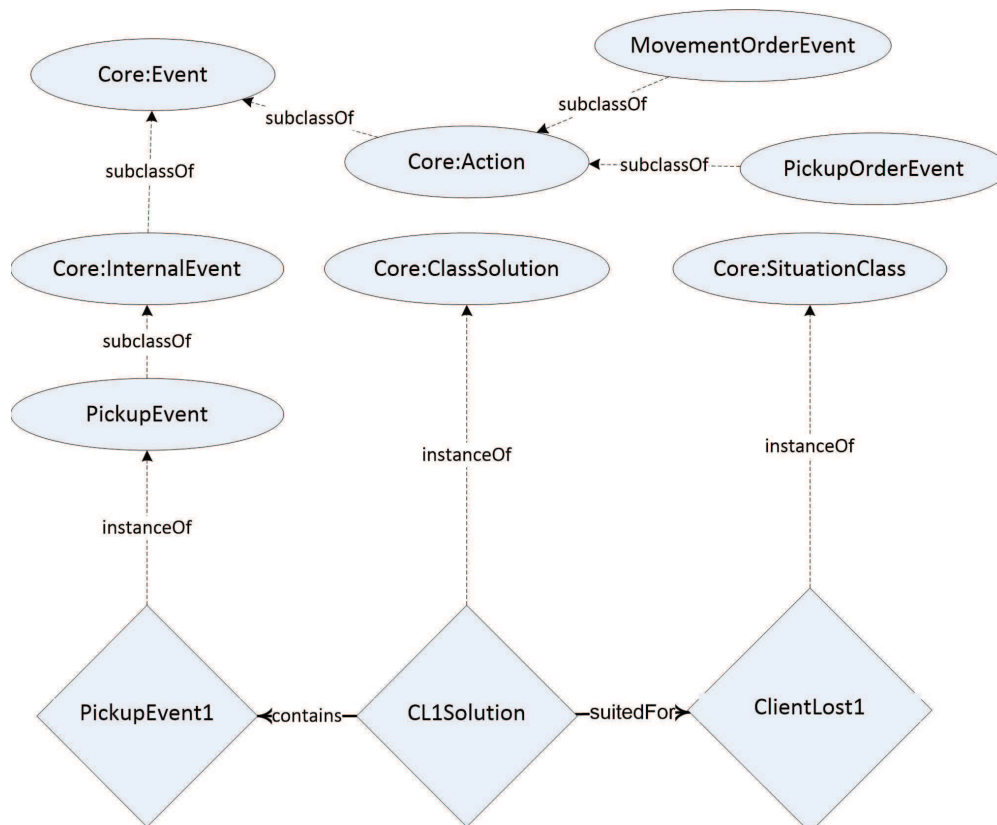


Рис. 4. Онтология описания тестовой ситуации

$\text{Taxi}(\text{?taxi1}) \wedge \text{MovementOrderEvent}(\text{?moveOrder}) \wedge$
 $\text{Interval}(\text{?tp1}) \wedge \text{holds}(\text{?moveOrder}, \text{?taxi1}) \wedge$
 $\text{holds}(\text{?moveOrder}, \text{?tp1}) \wedge (\text{?tp1}, \text{?orderReceived}) \Rightarrow$
 $\text{makeInstance}(\text{?moveEvent}) \wedge \text{makeInstance}(\text{?tp2}) \wedge$
 $\text{MovementEvent}(\text{?moveEvent}) \wedge \text{Interval}(\text{?tp2}) \wedge$
 $\text{holds}(\text{?moveEvent}, \text{?tp2}) \wedge$
 $\text{hasBeginning}(\text{?tp2}, \text{?orderReceived}).$

События `MovementOrderEvent` и `ExecutePickupOrderEvent` в описании предметной области наследуют понятию `Action` базовой онтологии, то есть являются результатом действий пользователя системы, лица, принимающего решения. В процессе обработки доступных решений машина вывода работает по принципу обратного вывода, проводя поиск по продукционным правилам, находящимся в онтологии. В данном случае в левой части правила-описания для события `PickupEvent`, помеченного как решение, содержатся события `VeryCloseSituatingEvent` и `ExecutePickupOrderEvent`. Второе из них является событием-действием, в отличие от первого, в левой части описания которого рекурсивно проводится поиск вхождения других событий. Процесс продолжается до тех пор, пока не будут обнаружены все события-действия (в данном случае `MovementOrderEvent` и `PickupOrderEvent`). Полу-

ченный в результате поиска список необходимых действий выдаётся пользователю в качестве одного из возможных решений для ситуации.

Заключение

Семантический подход к анализу ситуаций позволяет эксперту или группе экспертов описывать при помощи единого стандартизованного языка общую модель исследуемой предметной области, а также задавать с помощью используемого системой набора продукционных правил способ обработки исходных данных, поступающих в онтологическую модель в виде экземпляров понятий.

Применение семантических технологий в системах анализа ситуаций обеспечивает гибкость описания структуры модели объекта управления, а также гибкость описания его поведения в виде набора правил. Кроме этого, применение семантических технологий предоставляет экспертам, с одной стороны, возможность вносить в неё свои знания и, с другой стороны, сохраняет возможность автоматизированной обработки массивов данных нижнего уровня, поступающих с датчиков. Онтологическое описание предметной области и метода обработки исходных данных при помощи стандартизованных консорциумом W3C языков позволяет делать вывод о высокой масштабируемости подобного решения.

СПИСОК ЛИТЕРАТУРЫ

1. Поспелов Д.А. Ситуационное управление: Теория и практика. – М.: Наука, 1986. – 288 с.
2. Manola F., Miller E. RDF Primer. W3C Working Draft // World Wide Web Consortium (W3C). 2004. URL: <http://www.w3.org/TR/rdf-primer> (дата обращения: 09.09.2012).
3. Herman I., Horrocks I., Patel-Schneider P. F. et al. OWL 2 Web Ontology Language Document Overview // World Wide Web Consortium (W3C). 2007. URL: <http://www.w3.org/2004/OWL/> (дата обращения: 09.09.2012).
4. Matheus C., Kokar M. et al. SAWA: An Assistant for Higher-Level Fusion and Situation Awareness // SPIE Conference on Multisensor, Multisource Information Fusion: Proceedings of SPIE. – Orlando, FL., 2005. – V. 5813. – P. 75–85.
5. Patel-Schneider P.F., Horrocks I. et al. SWRL: A Semantic Web Rule Language Combining OWL and RuleML // World Wide Web Consortium (W3C). 2004. URL: <http://www.w3.org/Submission/SWRL/> (дата обращения: 09.09.2012).
6. Welty C., Hawke S. Rule Interchange Format // RIF Working Group. 2005. URL: http://www.w3.org/2005/rules/wiki/RIF_Working_Group (дата обращения: 09.09.2012).
7. Renz J. Qualitative Spatial Reasoning with Topological Information. Lecture Notes in Computer Science 2293. – Berlin: Springer Verlag, 2002. – 207 p.
8. Hobbs J., Pan F. Time Ontology in OWL // World Wide Web Consortium (W3C). 2006. URL: <http://www.w3.org/TR/owl-time/> (дата обращения: 09.09.2012).

Поступила 10.09.2012 г.