

УДК 004.825

СОБЫТИЙНЫЙ ПОДХОД К УПРАВЛЕНИЮ СЕМАНТИКОЙ СИСТЕМЫ

Починский И.А., Зинкин С.А.

ФГБОУ ВПО «Пензенский государственный университет», Пенза, e-mail: i.pochinskiy@yandex.ru

В статье указывается, что недостатки существующих средств представления знаний вносят семантическую несостоятельность и недетерминированность в модель системы. Приводятся свойства процедурных знаний, которые существующими средствами представления знаний не могут быть реализованы в полной мере. В качестве решения задачи поддержки свойств процедурных знаний вводится и формально описывается в терминах семантических гиперграфов понятие события в семантике системы. Событие рассматривается как формализм генерации фронта готовых продуктов системы продукционных правил. Также в статье предлагается расширение базового формального описания продукционных правил, благодаря которому появляется возможность учитывать их приоритет, исключительные ситуации, пред- и постусловия, а также возможность группировки правил. Выполнение с учетом приоритета правил из фронта готовых продуктов моделирует изменение семантики системы. При этом поддерживаются свойства связности, семантической метрики и активности процедурных знаний.

Ключевые слова: процедурные знания, продукционные правила, семантические гиперграфы, события, интеллектуальные системы управления знаниями (ИСУЗ), онтологии

EVENT-BASED SYSTEM SEMANTIC CONTROL

Pochinskiy I.A., Zinkin S.A.

Penza State University, Penza, e-mail: i.pochinskiy@yandex.ru

In the article there is noticed, that existing knowledge representation means disadvantages bring an inconsistency and indeterminacy in the system model. There are procedural knowledge properties, that couldn't be fully represented by existing knowledge representation means. Bringing solution of the procedural knowledge properties support problem there's considered the event in the system semantic terms. There's formal description of this term by the semantic hypergraph terms in the article. Event is considered as a ready production rules forefront of the production rules system generation formalism. Also in the article there's proposed advanced production rules formal description that allow taking into consideration rules' priorities, exceptions, pre conditions and post conditions and making rules groups. Prioritized executing ready production rules forefront models system semantic changes. Herewith procedural knowledge properties are supported.

Keywords: procedural knowledge, semantic hypergraph, production rules, events, intellectual knowledge management systems (IKMS), ontology

Описания декларативных знаний недостаточно для построения полнофункциональных интеллектуальных систем управления знаниями (ИСУЗ).

Есть свойства знаний [1] [5], которые невозможно [2] описать только с помощью языков представления декларативных знаний:

1. Связность (должна быть предусмотрена возможность установления новых связей различного типа между информационными единицами).

2. Семантическая метрика (на множестве информационных единиц в некоторых случаях полезно задавать отношение, характеризующее ситуационную близость информационных единиц, т.е. силу ассоциативной связи между информационными единицами).

3. Активность (не только появление в базе фактов или описаний каких-либо событий, но и установление новых связей и даже сам факт изменения знаний может стать источником активности ИСУЗ).

Для обеспечения возможности реализации этих свойств знаний необходимо использовать механизм вывода новых семантических зависимостей на основе уже известных фактов и набора правил.

Событие как формализм генерации фронта готовых продуктов

Изменение семантики моделируемой системы производится вследствие последовательности событий, происходящих в моделируемой системе, поэтому необходимо определить и формально описать понятие события.

Событием будем называть любое изменение состояния моделируемой системы, повлекшее модификацию семантической составляющей ее модели. Само событие удобно представлять как вершину семантического гиперграфа, обладающую рядом специализированных свойств, раскрывающих суть события. В терминах аппарата семантических гиперграфов формально событие можно описать как объект следующего класса:

$$S(event) = \left\{ \begin{array}{l} identified_with : id ; \\ evType : type ; \\ evID : evID ; \\ evName : name ; \\ evObject : object ; \\ evSubject : subject \end{array} \right.$$

Свойства *id* и *evID* используются для идентификации вершины: первое – глобально в пределах гиперграфа, второе – в пределах класса *event*.

Поскольку события предполагают модификацию исходных знаний, с каждым из них должен быть связан определенный набор правил, в соответствии с которыми производится изменение семантики модели системы. Модификация, которой подвергается семантический гиперграф в результате выполнения продукций, может быть произведена с помощью операций, формально описанных в [4].

Фактическим будем называть событие, значение всех атрибутов которого определены императивно (т.е. получены не в процессе моделирования, а были заданы на этапе проектирования).

Эвристическим будем называть событие, значение одного или нескольких атрибутов которого было получено в процессе моделирования (т.е. при трансформации гиперграфа вследствие возникновения других событий).

Ситуацией (событийной) будем называть совокупность событий, выделенную по каким-либо принципам. *Элементарной* будет являться ситуация, содержащая единственное событие. *Фактической* будем называть *ситуацию*, все события которой являются фактическими, а *эвристической* – все события которой являются эвристическими. Также можно определить понятие *глобальной и локальной ситуации*: совокупность всех событий системы и любое ее подмножество соответственно. Теоретически ситуации можно рассматривать как одно из средств задания коннотативной семантики системы или отдельных ее фрагментов.

Одно событие может являться частью нескольких ситуаций. Например, событие «Отключение электричества в одном доме» может входить в ситуацию «Возникновение аварии на сети доступа провайдера», в ситуацию «Сбой электропитания на подстанции» и в ситуацию «Поломка клиентского устройства доступа к сети». При использовании таких событий необходимо предусматривать корректную обработку последствий их возникновения в нескольких ситуациях.

Множество ситуаций модели системы можно рассматривать как множество наборов вариаций ее поведения, метазнания. Причем актуальность той или иной ситуации определяют события, входящие в нее. Поскольку, как было сказано выше, одно событие может быть частью нескольких ситуаций, в один момент времени могут быть актуальными несколько ситуаций. Здесь и далее подразумевается, что ситуация ак-

туальна, когда выполняется (значение слова «выполняется» применительно к продукциям объяснено ниже) хотя бы одна продукция из фронта готовых продукций одного или нескольких событий, являющихся частью этой ситуации.

При моделировании систем с заранее известным набором возможных событий (такие системы будем называть *фактическими* или *императивными*) вершины-экземпляры класса *event* создаются заранее, до запуска процесса моделирования. Во время моделирования при возникновении в системе одного из возможных событий запускается выполнение фронта готовых продукций, который в свою очередь модифицирует исходные декларативные знания.

В случае если набор возможных в системе событий не определен заранее (такие системы будем называть *эвристическими*), в модели должна быть реализована операция создания новых событий, а модель, соответственно, должна обладать более интеллектуальной подсистемой управления событиями, способной создавать новые, а также редактировать существующие продукты, расширяя их сферу применимости на созданное событие.

В таких системах при возникновении какого-либо события создается экземпляр класса *event*, заполняются значения его свойств (в т.ч. производится семантическое связывание вершин-свойств созданного экземпляра и вершин-экземпляров или вершин-классов, которые участвуют в событии), при необходимости создаются новые вершины-свойства, позволяющие глубже раскрыть его семантическую составляющую.

Множество вариантов поведения модели системы может варьироваться в зависимости от последовательности актуализации событийных ситуаций, причем эта последовательность может определяться как императивно (т.е. алгоритм поведения модели жестко задан при проектировании), так и эвристически. При эвристическом способе определения поведения модели системы последовательность событий может определяться по причинно-следственным (каузальным) связям между событиями (т.е. срабатывание определенного события может зависеть от факта срабатывания одного или нескольких других событий модели или от значения определенной вершины-свойства). Графически каузальные семантические дуги могут ничем не отличаться от обычных семантических дуг семантического гиперграфа.

Каузальным связыванием будем называть семантическое связывание каузальны-

ми связями. Формально каузальное связывание описывается следующим образом:

$$S(e_i) \leftarrow^+ \{ \text{causeArc} : \text{cause} \};$$

$$S(e_i :: \text{cause}) = e_j,$$

где e_i – событие, являющееся причиной; e_j – событие, являющееся следствием; causeArc – каузальная семантическая дуга.

При моделировании последовательной системы при возникновении в системе какого-либо события в модели системы формируется фронт готовых продукций, в который входят правила, характерные для возникшего события. Очередность выполнения этих продукций определяется приоритетом каждой из них (или известными технологиями управления очередями).

С одним событием может быть связано несколько продукционных правил. В этом случае при возникновении события все продукций, в сфере применимости которых указано именно оно, добавляются в фронт готовых продукций.

После выполнения всех правил фронта готовых продукций в системе происходит следующее событие, для моделирования которого формируется следующий фронт готовых продукций.

Формальное описание продукций в ретроспективе управляющих конструкций над семантическими гиперграфами

Необходимость расширения формальных способов описания семантики моделируемой системы упоминается в работе [3], также предлагается использовать в качестве формализма описания процедурных знаний системы продукционные правила.

При использовании классического определения продукций, приведенного в [1], с ростом количества продукций в модели системы увеличивается ее противоречивость, т.к. продукций могут конфликтовать друг с другом. Кроме того, система становится недетерминированной, т.к. не определена очередность выполнения продукций (в худшем случае некоторые продукций могут заблокировать выполнение других, модифицировав фрагмент гиперграфа, необходимый для удовлетворения условия их срабатывания). В связи с этим предлагается использовать следующее формальное определение продукций:

$$(i) = \langle S, L, A \rightarrow B_1 \otimes B_2, Q, Pr, Ex \rangle,$$

где i – имя или идентификатор продукционного правила. В качестве идентификатора продукций предлагается использовать ее порядковый номер записи в системе, перед

которым ставится символ #. Для решения вопроса увеличения удобочитаемости продукций человеком можно в памяти ЭВМ хранить и идентификатор, и название правила, используя для машинной обработки только идентификатор, а для отображения – название (или и то, и другое).

S – сфера применения продукции. Именно эта составляющая связывает события и правила модификации знаний, определяя фронт готовых продукций события. В сферу применения продукции может входить идентификатор события, тип событий (в этом случае продукция войдет во фронт готовых продукций при возникновении любого события данного типа) или их комбинации.

L – одно или несколько условий применимости ядра продукции. Может зависеть от значения какой-либо вершины-свойства, от наличия определенной семантической связи между несколькими вершинами, или в пределах всего гиперграфа, или от факта выполнения другой продукции.

$A \rightarrow B_1 \otimes B_2$ – ядро продукции в классическом его понимании: из истинности A следует истинность B_1 , из ложности A следует истинность B_2 . Другими словами, ЕСЛИ A , то B_1 , ИНАЧЕ B_2 . В качестве B_1 и B_2 могут использоваться как атомарные операции (например, удаление/создание вершины или семантической дуги), так и комплексные действия, модифицирующие большие фрагменты гиперграфа.

Q – постусловие продукции, которое актуализируется только после выполнения ее ядра (независимо от того, было выполнено действие B_1 или B_2).

Pr – приоритеты продукций, в соответствии с которым определяется очередность их выполнения. Приоритет может задаваться явно, либо определяться на основе правил. В случае если во фронт готовых продукций вошли несколько продукций, имеющих одинаковый приоритет, задача определения очередности должна быть решена разработчиком (при решении данной задачи можно использовать известные технологии управления очередями).

Ex – условие, при истинности которого ядро продукции активизировано не будет. Может являться конкретизацией условия применимости ядра продукции.

Благодаря наличию условия применимости и постусловия продукция становится возможным построение СУЗ таким образом, чтобы поддерживалось свойство активности знаний. Для этого результат выполнения одного или нескольких продукционных правил (или даже сам факт выполнения) должен инициировать выполнение одно-

го или нескольких других производственных правил.

Под выполнением продукции здесь и далее подразумевается последовательное выполнение следующих шагов:

1. Проверка условия срабатывания ядра продукции. Если условие выполняется, переход к шагу 2, иначе – к шагу 5.

2. Проверка условия-исключения. Если условие выполняется, переход к шагу 5, иначе – к шагу 3.

3. Активация ядра продукции.

4. Выполнение постусловия продукции.

5. Переход к следующей продукции из фронта готовых изделий.

Вывод

Событийное представление динамики семантики моделируемой системы представляет достаточно гибкий и выразительный инструмент представления процедурных знаний, хорошо описываемый в терминах семантических гиперграфов. С помощью такого подхода возможно ситуационное моделирование систем, поведение которых заранее не определено.

Список литературы

1. Искусственный интеллект. Справочник в трех томах / под ред. В.Н. Захарова, Э.В. Попова, Д.А. Поспелова, В.Ф. Хорошевского. – М.: Радио и связь, 1990.
2. Починский И.А. Проектирование онтологии сети передачи данных на основе формализма семантических гиперграфов // Труды Всероссийского конкурса научно-исследовательских работ студентов и аспирантов в области технических наук : материалы работ победителей и лауреатов конкурса. – СПб.: Изд-во Политехн. ун-та, 2012. – С. 139–141.
3. Починский И.А. Производственные правила в качестве средства формализации семантического гиперграфа // Уни-

верситетское образование: сб. статей XVI Междунар. научно-метод. конф. – Пенза: Изд-во ПГУ, 2012. – С. 443–445.

4. Починский И.А. Формальное представление семантических гиперграфов и операций над ними. // Молодежь. Наука. Инновации: Труды V междунар. научно-практ. интернет-конференции / под ред. Г.К. Сафаралиева, А.Н. Андреева, В.А. Казакова – Пенза: Изд-во Пензенского филиала ФГБОУ ВПО «РГУИТ», 2012. – С. 373–377.

5. Zadeh L. Commonsense knowledge representation based on fuzzy logic // Computer. – 1983. – Vol. 16. – С. 256–281

References

1. Zaharov V.N., Popov E.V., Pospelov D.A., Horoshevkiy V.F. *Iskusstvenny intellect [Artificial Intelligence]*, Moscow, Radio and connection, 1990.
2. Pochinskiy I.A. *Trudy Vserossijskogo konkursa nauchno-issledovatel'skih rabot studentov i aspirantov v oblasti tehniceskikh nauk: materialy rabot pobeditelej i laureatov konkursa (Proc. All-Russian competition of the science-research works of students and graduates in the technical science area: winners' and laureates' works)*. St-Petersburg, 2012, pp. 139–141.
3. Pochinskiy I.A. *Sbornik statej 16 mezhdunarodnoj nauchno-metodicheskoj konferencii (Works of 16th Int. Scientific and Methodic Conference)*, Penza, 2012, pp. 443–445.
4. Pochinskiy I.A. *Trudy 5 mezhdunarodnoj nauchno-prakticheskoj konferencii «Molodezh'. Nauka. Innovacii» (Proc. of the 5th Int. Science and Practic Conference «Youth. Science. Innovations»)*, Penza, 2012, pp. 373–377.
5. Zadeh L. Commonsense knowledge representation based on fuzzy logic // Computer. 1983. Vol. 16. pp. 256–281

Рецензенты:

Егоров С.И., д.т.н., доцент, профессор кафедры вычислительной техники, ФГБОУ «Юго-Западный государственный университет»;

Ромм Я.Е., д.т.н., профессор, заведующий кафедрой информатики, ФГБОУ ВПО «ТГПИ имени А.П. Чехова».

Работа поступила в редакцию 07.05.2013.