

# ИДЕНТИФИКАЦИЯ БИНАРНО ПОРОЖДАЕМЫХ СОБЫТИЙ ПРОЦЕССОВ ОБРАБОТКИ ЗНАНИЙ ДЛЯ СЕМАНТИЧЕСКОГО ПРОТОКОЛИРОВАНИЯ

Ивашенко В. П.

Кафедра интеллектуальных информационных технологий, Факультет информационных технологий и управления, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: ivashenko@bsuir.by

*В статье рассматриваются принципы и алгоритм идентификации событий для семантического протоколирования на основе бинарно порождаемых событий*

## ВВЕДЕНИЕ

Семантическое протоколирование (СемПр) – механизм, призванный обеспечить интеллектуальную систему интроспективными возможностями с целью её надления качествами искусственного сознания, включая возможность объяснять собственную работу и её результаты [1]. СемПр заключается в записи на языке представления знаний, например в виде семантической сети, знаний о порядке (системе) выполняемых действий и событий (явлений) в процессах обработки знаний. С точки зрения обработки и управления данными СемПр тесно связано с таким свойством структур данных (СД) как персистентность. С точки зрения же СемПр спецификой является то, что рассматривается персистентность не СД отдельного типа, но СД, обладающей семантикой языка представления знаний, отражающей всю базу знаний интеллектуальной системы, с учётом распределённого характера работы с ней. Различают следующие виды персистентности СД [2]: частичную, когда доступны последовательно предшествующие версии СД и только последняя из них (текущая) доступна для модификации; полную, когда любую из последовательно предшествующих версий СД можно модифицировать (ветвящееся в будущее время); конфлюэнтную персистентность (персистентность со слиянием), когда пространственно-временная структура процессов обработки СД является ориентированным ациклически графом (DAG), и одна версия СД может иметь более одной предшествующей. При обработке знаний, однако, возможны процессы и с более сложной структурой, например, циклические, что может быть выявлено или обусловлено необходимостью интеграции знаний. Кроме того, работа с семантическим протоколом предполагает решение и других задач, кроме задач получения версии СД: задачи определения какое из событий (явлений) является предшествующим, какие явления (события) предшествуют каждому из заданных явлений, итерирование по предшествующим событиям и др. Ранее были предложены СД [1] и алгоритмы для решения задач СемПр [3], например, такая предложенная

структура как триплетно-свободный биномиальный стек [1] и соответствующие алгоритмы работы [3] с ней способна обеспечить полную персистентность структуры протоколируемых событий (явлений), а «пирамидальная структура» [1], структуры и алгоритм для протоколирования независимых событий [1] ориентированы на поддержку персистентности со слиянием.

## ЗАДАЧИ СЕМАНТИЧЕСКОГО ПРОТОКОЛИРОВАНИЯ

В частном случае можно выделить процессы, в которых в (у)становлении каждого события участвуют не более пары событий. Такие события будем называть бинарно порождаемыми. В самом начале протокола находятся первичные события, которые не имеют устанавливающих событий. Каждое из последующих событий имеет двух предков, порождающих данное событие. Для каждого события может быть измерена длина – расстояние от самого дальнего первичного события по цепочке становлений. Событие и порождающие его события могут быть ассоциированы в грань в виде треугольника, в верхнем углу которого находится порождённое событие, а в углах при основании находятся порождающие события. Иерархическое продолжение построения ассоциаций в виде треугольных граней (получаемые грани аналогичным образом ассоциируются в грани более высокого уровня) позволяет получить пирамидальную структуру, в основании которой будет находиться структура, напоминающая структуру треугольника Серпинского. При конечном числе событий такая структура конечна. Грань верхнего уровня такой структуры будем называть вершиной (вершиной пирамидальной структуры), а количество уровней, на которых располагаются грани, будем называть высотой пирамидальной структуры. Для того чтобы различать события, с каждым событием, кроме вершины пирамидальной структуры, в которой находятся все события, предшествующие описываемому событию, ассоциируется некоторый идентификатор.

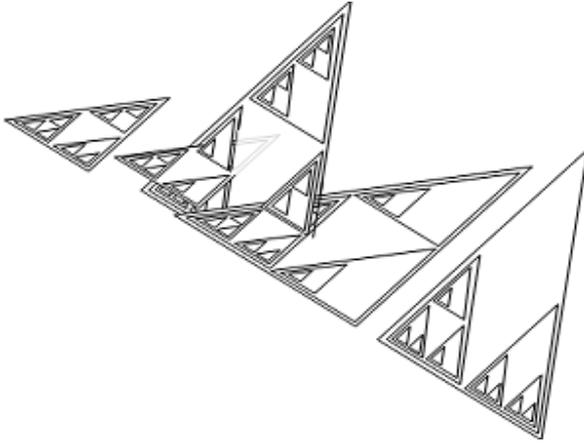


Рис. 1 – Схематическое представление семантического протокола

Особенностью подхода является то, что идентификатор каждого события (*EID*) формируется динамически по мере построения протокола в виде пирамидальных структур (см. Рис. 1), поэтому количество бит в идентификаторе (*Capacity*) пропорционально высоте сформированной пирамидальной структуры и называется «высотой идентификатора». Комбинированный идентификатор (*CID*) является результатом комбинации (объединения) идентификаторов ( $k, i$ ) ассоциируемых гранью граней или событий (см. *combineCID* (1)). Так как каждый *EID* представляется цепочкой, то *CID* является структурой, соответствующей префиксному дереву идентификаторов событий, частью механизма индексации при поиске вхождения события в пирамидальную структуру.

```

combineCID
⟨c, k, i⟩ ← /* c – context (knowledge base) */
if ( $i \neq \emptyset$ )
  if ( $k = \emptyset$ )  $k \leftarrow i$ 
  else if (joined(⟨c, k, i⟩))
    |⟨c, k⟩ ← recombineCID(⟨c, k, i⟩)
  else
    |⟨c, t⟩ ← getCapacity(⟨c, i⟩)
    |⟨c, d⟩ ← getCapacity(⟨c, k⟩)
    |⟨c, f⟩ ← getFreeCapacity(⟨c, i⟩)
    |⟨c, b⟩ ← getFreeCapacity(⟨c, k⟩)
    |if ( $t \leq b$ ) ⟨c, k⟩ ← joinCID(⟨c, i, k, t, b⟩)
    |else
      |if ( $d \leq f$ ) ⟨c, k⟩ ← joinCID(⟨c, k, i, d, f⟩)
      |else
        |while ( $t - - > d$ )
          |⟨c, i⟩ ← setLower(⟨c, \emptyset, i⟩)
          |while ( $- - d > t$ )
            |⟨c, k⟩ ← setLower(⟨c, \emptyset, k⟩)
            |⟨c, k⟩ ← combineNode(⟨c, \emptyset, i, k⟩)
        |⟨c, k⟩
  |⟨c, k⟩

```

(1)

С каждой вершиной в дереве ассоциируется максимальная высота свободной ветви (*FreeCapacity*) в поддереве. При объединении деревьев, если высота одного из деревьев не превосходит максимальной высоты свободной вет-

ви в другом дереве, то это дерево помещается в качестве поддерева в узел ветви, которая становится занятой, максимальные высоты для узлов на пути от неё выше к корню соответственно обновляются. Иначе объединённое дерево строится путём добавления в качестве корня новой вершины, а в качестве поддеревьев в него добавляются объединяемые деревья (см. *joinCID* (2)).

```

joinCID
⟨c, i, k, t, b⟩ ←
⟨n, u⟩ ← ⟨k, k⟩
while ( $t \leq b$ )
  |⟨f, p⟩ ← ⟨b, n⟩
  |⟨c, l⟩ ← getLower(⟨c, p⟩)
  |⟨c, n⟩ ← getUpper(⟨c, p⟩)
  |if ( $\emptyset \neq l$ )
    |⟨c, b⟩ ← getFreeCapacity(⟨c, l⟩)
    |if ( $t \leq b$ )  $n \leftarrow l$ 
    |⟨c, b⟩ ← getFreeCapacity(⟨c, n⟩)
    |if ( $(b < f) \wedge (t \leq b)$ )  $u \leftarrow p$ 
  |while ( $t + + < b$ ) ⟨c, i⟩ ← setLower(⟨c, \emptyset, i⟩)
  |⟨c, l⟩ ← getLower(⟨c, p⟩)
  |if ( $l = \emptyset$ ) ⟨c, p⟩ ← setLower(⟨c, i, p⟩)
  |else ⟨c, p⟩ ← setUpper(⟨c, i, p⟩)
  |while ( $u \neq p$ )
    | $n \leftarrow u$ 
    |⟨c, l⟩ ← getLower(⟨c, n⟩)
    |⟨c, u⟩ ← getUpper(⟨c, n⟩)
    |if ( $\emptyset \neq l$ )
      |⟨c, b⟩ ← getFreeCapacity(⟨c, l⟩)
      |if ( $t \leq b$ )  $u \leftarrow l$ 
      |⟨c, u⟩ ← setFreeCapacity(⟨c, b, u⟩)
    |⟨c, k⟩

```

(2)

## ЗАКЛЮЧЕНИЕ

Разработанные алгоритмы применимы для протоколирования процессов поддержки проектирования баз знаний, решения задач, включая обработку образов, в совместимых с технологией OSTIS [4] интеллектуальных системах.

1. Ивашенко, В. П. Алгоритмы семантического протоколирования процессов обработки знаний / В. П. Ивашенко // BIG DATA Advanced Analytics: collection of materials of the fourth international scientific and practical conference, Minsk, Belarus, May 3–4, 2018 / editorial board: M. Batura [et al.]. – Minsk, BSUIR, 2018. – P. 267–273.
2. Driscoll, J. R. Fully Persistent Lists with Catenation / J. R. Driscoll, D. D. Sleator, R. E. Tarjan. – 1994.
3. Ивашенко, В. П. Семантическое протоколирование процессов обработки знаний / В. П. Ивашенко // Информационные технологии и системы 2017 (ITS 2017) = Information Technologies and Systems 2017 (ITS 2017) : материалы междунар. науч. конф. (Республика Беларусь, Минск, 25 октября 2017 года) / редкол. : Л. Ю. Шилин [и др.]. – Минск : БГУИР, 2017. – С. 110–111.
4. Голенков, В. В. Семантическая технология компонентного проектирования систем, управляемых знаниями / В. В. Голенков, Н. А. Гулякина // OSTIS. 2015. — С. 57—78.