

ИСПОЛЬЗОВАНИЕ МЕТАГРАФОВ ДЛЯ ОПИСАНИЯ СЕМАНТИКИ И ПРАГМАТИКИ ИНФОРМАЦИОННЫХ СИСТЕМ

Э.Н. Самохвалов, Г.И. Ревунков, Ю.Е. Гапанюк

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

e-mail: eduard.samohvalov@yandex.ru; revunkov@bmstu.ru; gapyu@bmstu.ru

Предложен подход к описанию семантики и прагматики информационных систем с использованием метаграфов, понятие семантики информационной системы как графовой структуры для описания данных и знаний, способ представления и хранения метаграфов в виде информационного элемента метаграфа, а также основные операции над метаграфами, которые могут быть реализованы в виде программного интерфейса. Показано развитие понятия семантики информационной системы. Дано определение метаграфа и формализованное описание его информационного элемента. На примере показаны различия между гиперграфом и метаграфом. Рассмотрены основные виды атрибутов информационного элемента метаграфа, описание целей информационной системы в виде частично-упорядоченного множества, пример описания данных и процессов информационной системы с использованием метаграфовой структуры. Предложено формализованное описание метаграфового агента в виде реактивного агента, реализующего поведение на основе заданного набора правил, описание прагматики информационной системы в виде связи частично-упорядоченного множества целей с процессами на основе метаребер метаграфа.

Ключевые слова: семантика информационной системы, прагматика информационной системы, метаграф, гиперграф, информационный элемент метаграфа, метавершина, метаребро, метаграфовый агент.

METAGRAPHS FOR INFORMATION SYSTEMS SEMANTICS AND PRAGMATICS DEFINITION

E.N. Samokhvalov, G.I. Revunkov, Yu.E. Gapanyuk

Bauman Moscow State Technical University, Moscow, Russian Federation

e-mail: eduard.samohvalov@yandex.ru; revunkov@bmstu.ru; gapyu@bmstu.ru

The authors propose approach: for information systems semantics and pragmatics definition using metagraphs, concept of information systems semantics in form of graph structure for data and knowledge definition, method of representation and storage of metagraphs based on metagraph information element (MIE), as well as the basic operations over metagraphs, which could be implemented as a program interface. Evolution of concept of information systems semantics is shown. Definition of metagraph and formalized description of its information element is given. Differences between hypergraph and metagraph are shown by the example. Basic kinds of MIE attributes are analyzed. Definition of the goals of the information system in the form of a partially ordered set is considered. Using metagraph structures the example of data and processes definition of information system is examined. The authors have developed: formalized model of metagraph agent as a reactive agent that implements the behavior on the basis of a given set of rules; the definition of information system pragmatics in the form of communication of partially ordered set of goals with processes based on metaedges of metagraph.

Keywords: information systems semantics, information system pragmatics, metagraph, hypergraph, metagraph information element, metavertex, metaedge, metagraph agent.

Введение. Во второй половине прошлого века в рамках отечественной школы ситуационного управления был предложен подход к построению информационных систем на основе прикладной семиотики. Работы в этой области связаны, прежде всего, с именами Ю.И. Клыкова [1, 2] и Д.А. Поспелова [3, 4]. В частности в работе [1] отмечено, что “принципиальное отличие семиотических операций от формальных состоит в том, что в них отражается не только синтаксис, но также семантика и прагматика решаемых задач”.

Гибкость семиотического подхода по сравнению с формальным Клыков объясняет, в частности, тем, что семиотический подход рассматривает “класс формально-логических систем с переменной структурой. В этой модели переменный характер имеют как аксиомы, так и правила вывода. . . . Современная математика, как известно, имеет дело со статическими формальными системами. . . ” [1].

В рамках семиотического подхода информационную систему можно разделить на пассивную (семантика системы) и активную (прагматика системы) части.

Семантика представляет собой базу данных и знаний, необходимых для функционирования системы, а прагматика — набор действий, процессов, которые могут выполняться в системе. К прагматике также относятся цели системы, которые определяют требуемые наборы знаний и действий.

В результате анализа источников можно сделать вывод о том, что исторически существовало два подхода к понятию семантики информационной системы, подход I постепенно трансформировался в подход II.

Подход I. Семантика как отношение или соответствие. Это классический взгляд на семантику, который пришел в прикладную семиотику из общей семиотики. В частности, в работе [5] отмечено, что в соответствии с взглядами одного из основателей общей семиотики Ч. Морриса “семантика задает отношения между знаками и тем, что они обозначают”.

В рамках прикладной семиотики Клыков отмечает, что “семантикой знака являются правила, с помощью которых устанавливается соответствие между объектами модели проблемной среды и объектами реальной среды. Это соответствие имеет целенаправленный характер и устанавливается в процессе обучения . . . ” [1].

Однако постепенно стало понятно, что ключевым элементом, обеспечивающим соответствие, является структура данных, которая позволяет хранить многообразие отношений, реализующих такое соответствие. Так появился подход II.

Подход II. Семантика как структура данных и знаний. В современных работах (в частности, в области семантического веба) семантика понимается, прежде всего, как сложная структура, предназначенная для хранения данных и знаний. В настоящее время не вызывает сомнения, что для представления семантики наиболее подходящей является разновидность графовой модели. В рамках семантического веба разработаны стандарты RDF (универсальная графовая модель представления данных и знаний на основе триплетов), RDFS и OWL (модели представления онтологий, OWL – расширение стандарта RDFS). Таким образом, будем рассматривать семантику информационной системы как универсальную графовую структуру данных, которая позволяет описывать данные и знания информационной системы. Будет показано, что такая структура может быть использована для представления знаний о процессах информационной системы, т.е. для описания ее прагматики.

В настоящее время наиболее известной из таких структур является гиперграф [6, 10]. Однако в настоящей статье в качестве подобной структуры предлагается использовать метаграф.

Цель работы – рассмотрение преимуществ и особенностей использования метаграфов для представления семантики и прагматики информационной системы. Для выполнения указанной цели необходимо решить следующие задачи:

- рассмотреть формализованное описание метаграфа;
- показать преимущества метаграфов перед гиперграфами для решения поставленной задачи;
- предложить способ представления и хранения метаграфов в памяти информационной системы (для решения рассматриваемой задачи авторами предлагается модель информационного элемента метаграфа (ИЭМ));
- выявить состав основных операций над метаграфами в виде ИЭМ;
- предложить способ преобразования метаграфов в виде ИЭМ;
- разработать способ описания прагматики информационной системы с использованием метаграфов.

Графы и метаграфы. Одна из наиболее развитых графовых структур, которая может быть применена для представления семантики информационных систем, – гиперграф. В соответствии с работой [6] гиперграф определяется следующим образом: $HG = \langle \{hv_i\}, \{he_j\} \rangle$, $he_j = \{hv_i\}$, где hv_i – вершина гиперграфа; he_j – гиперребро гиперграфа.

Отличие гиперграфа от обычного графа заключается в том, что ребро гиперграфа может соединять произвольное множество вершин.

Если для ребра задан порядок обхода вершин, то гиперграф называют ориентированным.

В настоящее время гиперграфы широко используются в тех областях, где требуется анализировать сложные связи между вершинами графа, в частности, при проектировании аппаратного обеспечения [6].

Однако в силу определения гиперграф не обладает достаточно удобными возможностями моделирования иерархий. В противоположность гиперграфу, метаграф в соответствии с работой [7] “представляет собой модель иерархического объекта с различными уровнями обобщения”.

Основополагающие работы по теории метаграфов — работы А. Базу и Р. Блэннинга, которые в 2007 г. были обобщены в монографию [8]. Из отечественных работ хотелось бы отметить работу [7], в которой рассмотрены особенности применения метаграфов для описания моделей сложных объектов. В работе [7] указано, что единая теория метаграфов в настоящее время не сформирована. Поэтому в разных источниках можно встретить различные определения метаграфа, которые отличаются деталями.

В этой статье использовано определение, которое достаточно близко к определению Базу и Блэннинга, но адаптировано для описания семантики и прагматики информационной системы: $MG = \langle V, MV, E, ME \rangle$, где MG — метаграф; V — множество вершин метаграфа; MV — множество метавершин метаграфа; E — множество ребер метаграфа; ME — множество метаребер метаграфа.

Понятие метаребра отсутствует в классическом определении метаграфа, однако оно потребуется для описания прагматики информационной системы.

В работе [8] вводится понятие атрибутивного метаграфа MGA , в котором каждой вершине и ребру может быть приписано произвольное число атрибутов — числовых, строковых и др. В настоящей статье использован метаграф в атрибутивной форме.

Вершина метаграфа характеризуется множеством атрибутов $v_i = \{atr_k\}$, $v_i \in V$, где v_i — вершина метаграфа; atr_k — атрибут.

Ребро метаграфа характеризуется множеством атрибутов, исходной и конечной вершиной и признаком направленности: $e_i = \langle v_S, v_E, eo, \{atr_k\} \rangle$, $e_i \in E$, $eo = true \mid false$, где e_i — ребро метаграфа; v_S — исходная вершина (метавершина) ребра; v_E — конечная вершина (метавершина) ребра; eo — признак направленности ребра ($eo = true$ — направленное ребро, $eo = false$ — ненаправленное ребро). Таким образом, ребро метаграфа соединяет две вершины в отличие от гиперребра гиперграфа, которое может охватывать множество вершин. На число ребер между двумя вершинами не накладывается ограничений, ребра могут различаться набором атрибутов, следовательно, метаграф

является разновидностью мультиграфа. Ребро может соединять как вершины, так и метавершины метаграфа.

Введем понятие фрагмента метаграфа $MG_i = \{ev_j\}$, $ev_j \in (V \cup E \cup MV \cup ME)$, где MG_i — фрагмент метаграфа; ev_j — элемент, принадлежащий объединению множеств вершин (метавершин) и ребер (метаребер) метаграфа. Таким образом, фрагмент метаграфа в общем виде может содержать произвольные вершины (метавершины) и ребра (метаребра) без ограничений. Ограничения вводятся на фрагменты метаграфа, входящие в метавершину и метаребро.

Метавершину метаграфа mv_i определим следующим образом: $mv_i = \langle \{atr_k\}, \{ev_j\} \rangle$, $mv_i \in MV$, $ev_j \in (V \cup E^{eo=false} \cup MV \cup ME^{eo=false})$. Таким образом, метавершина в дополнение к свойствам вершины включает в себя вложенный фрагмент метаграфа. При этом ребра и метаребра этого фрагмента могут быть только ненаправленными, $eo = false$.

Метаребро me_i метаграфа $me_i = \langle v_S, v_E, eo, \{atr_k\}, \{ev_j\} \rangle$, $e_i \in E$, $eo = true \mid false$, $ev_j \in (V \cup E^{eo=true} \cup MV \cup ME^{eo=true})$. Следовательно, метаребро в дополнение к свойствам ребра содержит вложенный фрагмент метаграфа. При этом ребра и метаребра этого фрагмента могут быть только направленными, $eo = true$.

Определения метавершины и метаребра рекурсивны, так как элементы ev_j могут быть метавершинами и метаребрами.

Наличие у метавершин собственных атрибутов и связей с другими вершинами является важной особенностью метаграфов. Это соответствует принципу эмерджентности, т.е. приданию понятию нового качества, несводимости понятия к сумме его составных частей. Фактически, как только вводится новое понятие в виде метавершины, оно “получает право” на собственные свойства, связи и т.д., поскольку в соответствии с принципом эмерджентности новое понятие обладает новым качеством и не может быть сведено к подграфу базовых понятий.

Отметим, что между метаграфом и гиперграфом существуют принципиальные различия. Гиперграф был придуман как формализм, позволяющий моделировать сложный порядок обхода вершин графа. Этот подход, в частности, нашел применение при конструировании электронных микросхем. Главная задача метаграфов — моделирование сложных иерархических объектов и систем. Схематическое изображение метаграфа и гиперграфа приведено на рис. 1.

Сплошными линиями на рис. 1 обозначены вершины и гипервершины, штриховыми — ребра и гиперребра. Ребра и гиперребра могут быть как ненаправленными (показано на рисунке), так и направленными. Гиперребро гиперграфа задает последовательность обхода вершин. Гиперребро he_1 включает в себя вершины hv_1, hv_3, hv_4 (см. рис. 1); а гиперребро he_2 — вершины hv_1 и hv_2 .

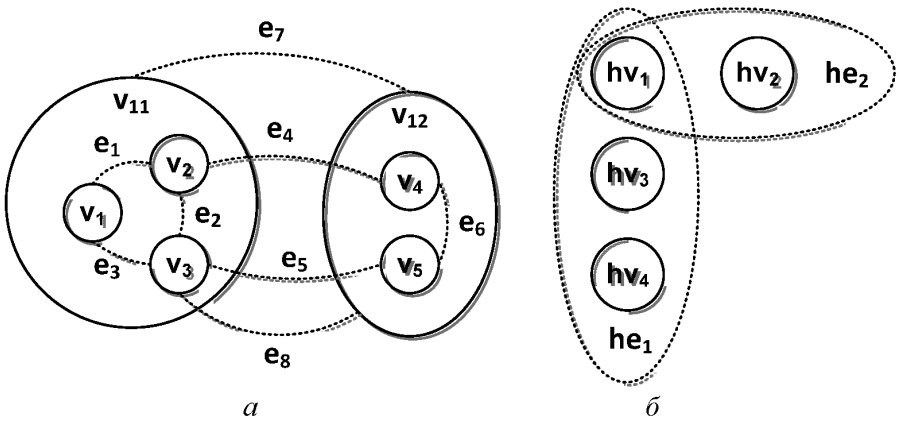


Рис. 1. Схематические изображения метаграфа (а) и гиперграфа (б)

Метаграф предназначен для моделирования сложных иерархических взаимосвязей между объектами. Две метавершины v_{11} (включает вершины v_1, v_2, v_3 и ребра e_1, e_2, e_3) и v_{12} (которая включает вершины v_4, v_5 и ребро e_6) показаны на рис. 1.

Если гиперребро гиперграфа может включать в себя только вершины, то метавершина метаграфа может содержать как вершины (или метавершины), так и ребра. Ребро метаграфа может соединять вершины внутри одной метавершины (e_1, e_2, e_3, e_6), вершины между различными метавершинами (e_4, e_5), метавершины (e_7), вершины и метавершины (e_8).

Рассмотренный пример показывает, что в отличие от гиперграфа, метаграф позволяет моделировать сложные иерархические системы.

Способ представления и хранения метаграфов. Определение метаграфа не дает полной информации о том, каким образом хранить метаграфы в информационной системе. Для ответа на этот вопрос рассмотрим представление метавершины в виде графа (рис. 2).

Метавершину (см. рис. 2) можно представить в виде комбинации обычной вершины и набора связей от этой вершины к другим вершинам (или метавершинам) и ребрам. Поскольку метавершина содержит связи не только с другими вершинами, но и с ребрами, то в этом слу-

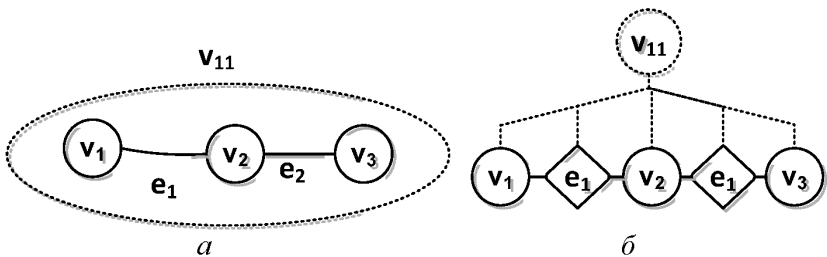


Рис. 2. Метавершина (штриховая линия) в виде фрагмента метаграфа (а) и ее представление в виде графа (б)

чае ребро удобно рассматривать в виде вершины отдельного класса, которая имеет неименованные связи с вершинами (метавершинами). Таким образом, вершины и ребра образуют двудольный граф с той поправкой, что если вершина является метавершиной, то она содержит дополнительные иерархические связи с другими вершинами (метавершинами) и ребрами.

Если метаграф аннотирован, то аннотации могут быть заданы для вершин (метавершин) и ребер, которые показаны на рис. 2, б окружностями и ромбами, но не могут быть заданы для “примитивных” связей, показанных на рис. 2, б линиями. В соответствии с графом, приведенным на рис. 2, б, можно сделать вывод о том, что СУБД на основе графовой модели могут оказаться неудобными для хранения метаграфов, так как в графовых СУБД концепции вершины и ребра четко отделены друг от друга. В частности, одна из наиболее зрелых графовых СУБД Neo4J, которая обладает развитым языком запросов, не позволяет провести связь от вершины к ребру, что фактически делает невозможным моделирование метавершин в рамках этой СУБД.

Для хранения метаграфов в большей степени могут подойти документо-ориентированная (на основе форматов XML или Json) или реляционная СУБД. Но для использования таких СУБД необходимо четко определить структуру вершины (метавершины) и ребра метаграфа. Следовательно, в случае представления вершин (метавершин) и ребер с помощью двудольного графа, вершины и ребра обладают относительно похожими структурами.

В настоящей работе предложено моделировать похожие структуры в виде единой структуры данных. Назовем такую единую структуру данных информационным элементом метаграфа, который является элементарной строительной единицей для конструирования метаграфов, описывающих семантику информационной системы.

Формализованная модель ИЭМ. Рассмотрим формализованную модель предлагаемого ИЭМ. Представим ИЭМ в виде следующего кортежа:

$$\text{ИЭМ} = \langle \text{id}, \text{NM}, \text{VAL}, \text{RL}, \{\text{lnk}_i\}, \{\text{atr}_j\} \rangle, \text{RL} \in \{\text{RL}_V, \text{RL}_{MV}, \text{RL}_R\},$$

где id — уникальный идентификатор элемента; NM — наименование элемента; VAL — значение элемента; RL — роль элемента; lnk_i — ссылка на другой ИЭМ; atr_j — атрибут; RL_V — роль элемента “вершина”; RL_{MV} — роль элемента “метавершина”; RL_R — роль элемента “ребро”. Таким образом, ИЭМ может использоваться для хранения вершины, метавершины и ребра метаграфа. С использованием ссылок lnk_i реализуются нетипизированные связи между вершиной и ребром, а также связи между метавершиной и входящими в нее элементами.

С учетом предложенной модели ИЭМ можно определить метаграф как множество входящих в него ИЭМ: $\text{MG} = \{\text{ИЭМ}_i\}$.

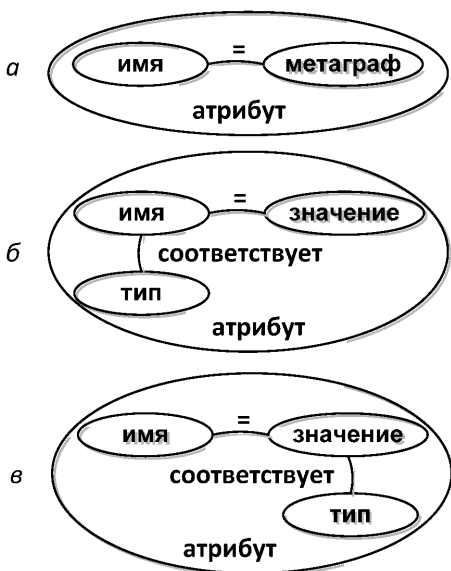


Рис. 3. Структуры ссылочного (а), типизированного (б) и нетипизированного (в) атрибутов

Атрибут в предлагаемой модели — метавершина метаграфа, может принадлежать одному из трех видов:

$$atr_j \stackrel{\text{def}}{=} MG, atr_j \in$$

$$\in \{ATRTR_{REF}, ATRRTT, ATRTNT\},$$

где $ATRTR_{REF}$, $ATRRTT$, $ATRTNT$ — ссылочный, типизированный и нетипизированный атрибуты.

Детализированные структуры атрибутов трех видов представлены на рис. 3. Ссылочный атрибут используется для ссылок на произвольный фрагмент метаграфа. Типизированный атрибут содержит имя, значение и тип данных. Тип данных приписан имени атрибута, атрибут может содержать

только значения заданного типа. Нетипизированный атрибут также содержит имя, значение и тип данных. Тип данных приписан значению атрибута, атрибуту можно присваивать значения различных типов. Тип данных атрибута изменяется в зависимости от типа присвоенного значения.

Множество типов атрибутов может быть задано в зависимости от потребностей предметной области. Атрибуты позволяют аннотировать элементы метаграфа текстовой и числовой информацией, а также информацией других типов. Поскольку атрибут является метавершиной, при необходимости он может быть аннотирован другими метавершинами.

Основные операции над метаграфами. Для динамической работы с метаграфами введем следующие основные операции над метаграфами и их элементами OP^{MG} .

1. Создание нового ИЭМ — формируется новый ИЭМ с заданными параметрами, уникальный идентификатор id генерируется автоматически: $ИЭМ = ИЭМ (\langle NM, VAL, RL, \{lnk_i\}, \{atr_j\} \rangle)$.

2. Добавление ссылки на ИЭМ — к информационному элементу $ИЭМ_i$ добавляется ссылка на информационный элемент $ИЭМ_j$: $ИЭМ_i = ИЭМ_i + lnk_j(ИЭМ_j)$.

3. Удаление ссылки на ИЭМ — у информационного элемента $ИЭМ_i$ удаляется ссылка на информационный элемент $ИЭМ_j$: $ИЭМ_i = ИЭМ_i - lnk_j(ИЭМ_j)$.

4. Добавление атрибута — к информационному элементу ИЭМ_i добавляется атрибут atr_j : $ИЭМ_i = ИЭМ_i + atr_j$.

5. Удаление атрибута — у информационного элемента ИЭМ_i удаляется атрибут atr_j : $ИЭМ_i = ИЭМ_i - atr_j$.

6. Добавление информационного элемента к метаграфу — к метаграфу добавляется ИЭМ_i: $MG_2 = MG_1 + ИЭМ_i$.

7. Удаление информационного элемента из метаграфа: $MG_2 = MG_1 - ИЭМ_i$.

8. Создание переменной-ссылки на фрагмент метаграфа — формирование переменной, которая содержит ссылку на фрагмент метаграфа: $VAR_i = ref(MG_j)$.

9. Создание переменной-копии фрагмента метаграфа — формируется переменная, которая содержит копию фрагмента метаграфа: $VAR_i = copy(MG_j)$.

10. Поиск ИЭМ по заданным параметрам — формируется метаграф, который представляет собой множество найденных ИЭМ: $MG_i = find(\langle id, NM, VAL, RL, \{lnk_i\}, \{atr_j\} \rangle)$.

В качестве параметров поиска могут использоваться любые параметры ИЭМ.

11. Оператор цикла $foreach(MG_i) \rightarrow \{OP_j^{MG}\}$. В цикле для каждого ИЭМ, входящего в состав метаграфа MG_i , выполняется заданное множество действий OP_j^{MG} . Для связи текущего обрабатываемого ИЭМ с выполняемыми в цикле действиями могут использоваться переменные.

12. Оператор создания функции $function(\{FP_i\}) \rightarrow \{OP_j^{MG}\}$. Функция с заданным множеством параметров FP_i позволяет сгруппировать множество операторов OP_j^{MG} . Параметры функции могут использоваться в качестве параметров операторов OP_j^{MG} . Например, можно рассмотреть функцию, которая принимает на вход элементы-вершины и формирует элемент-связь между этими вершинами.

Предложенные операторы могут быть достаточно просто реализованы в виде API (программного интерфейса) практически в любом современном языке программирования. При этом можно использовать уже готовые языковые механизмы работы с переменными и создания функций.

Использование метаграфов для описания прагматики информационной системы. Классическое определение структуры целей — представление целей в виде графовой структуры:

$$GT = \langle GN, \prec \rangle, GN = \{g_i\}, \quad (1)$$

где GT — структура целей; GN — множество целей; \prec — отношение порядка на множестве целей; g_i — i -я цель.

Часто под отношением порядка \prec понимается возможность задания целей в виде дерева. Однако более корректно понимать отношение

порядка \prec как отношение частичного порядка на множестве целей, так как выполнение одной цели нижнего уровня может требоваться для реализации нескольких целей верхнего уровня. Такое определение целей не дает никакой информации о способах достижения целей. Для задания подобной информации будем использовать концепцию метаграфов.

Рассмотрим, как предложенная формализованная модель метаграфа позволяет определить прагматику информационной системы. Для описания данных информационной системы используются метавершины, которые позволяют определять ненаправленные семантические связи между элементами данных. Наличие вложенности, которая обеспечивается метавершинами, дает возможность моделировать сложные иерархические отношения между элементами данных.

Для описания действий информационной системы используются метаребра, которые позволяют определять направленные процессы в информационной системе. Пример описания фрагмента прагматики информационной системы приведен на рис. 4. Окружностями показаны вершины и метавершины, используемые для описания данных, прямоугольниками — метавершины или метаребра, соответствующие элементам процесса, двойными прямоугольниками — метаребра, со-

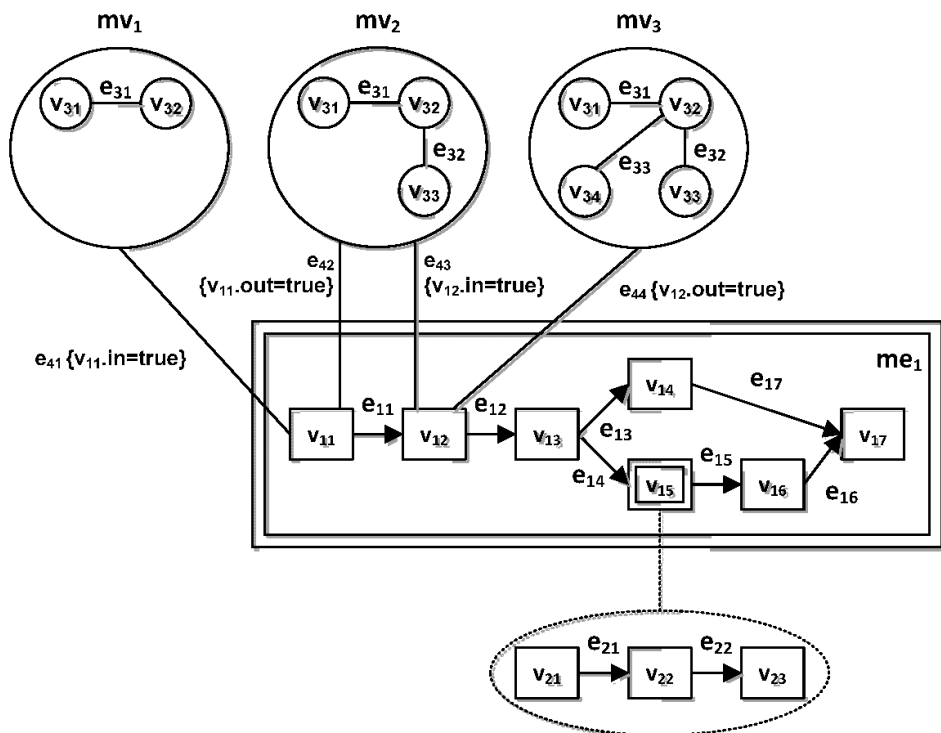


Рис. 4. Пример описания фрагмента прагматики информационной системы в виде фрагмента метаграфа

ответствующие описаниям процесса, ненаправленными связями – ненаправленные ребра ($eo = \text{false}$), а направленными стрелками – направленные ребра ($eo = \text{true}$), штриховой линией – вложенность фрагмента метаграфа.

Фрагмент прагматики информационной системы изображен в виде процесса (см. рис. 4), показанного метаребром me_1 . Процесс состоит из элементов процесса v_{1*} (под знаком «*» понимается произвольное значение второго индекса). Элементы процесса соединены направленными ребрами e_{1*} . В качестве примера для элемента процесса v_{15} показаны элементы вложенного подпроцесса. Таким образом, элемент v_{15} одновременно является и элементом процесса me_1 и метаребром, которое содержит вложенный процесс.

Данные, поступающие на вход элемента v_{11} , представлены метавершиной mv_1 , которая содержит вложенные вершины данных v_{31} и v_{32} и ненаправленную связь e_{31} между ними. Связь метавершины mv_1 с элементом процесса v_{11} осуществляется с помощью ребра e_{41} . Признаком того, что метавершина mv_1 содержит входные данные процесса v_{11} , моделируется с помощью атрибута ребра e_{41} (вершины и ребра могут иметь атрибуты, так как используется модель атрибутивного метаграфа). В рассматриваемом случае применяется атрибут $v_{11}.in = \text{true}$. Аналогично с помощью ребер e_{42} , e_{43} , e_{44} проводится привязка метавершин mv_2 и mv_3 к элементам процесса v_{11} и v_{12} в качестве входных–выходных данных. Случай, когда выходные данные предыдущего процесса являются входными данными следующего процесса, показан на рис. 4. Такой подход используется в некоторых моделях описания процессов, например в синергетических интерпрезентативных сетях (Synergetic Inter-Representation Network), предложенных Дж. Португали [11]. Однако предлагаемая модель связи метавершин данных с элементами процесса носит более гибкий характер и позволяет моделировать передачу данных как через входные–выходные метавершины данных, так и другими способами, например, через представленный метавершиной общий контекст.

Определим прагматику информационной системы $PR = \langle GT, GA \rangle$, $GA = \{ga_i\}$, $ga_i = \langle g_j, me_k \rangle$, где PR – прагматика информационной системы; GT – структура целей в соответствии с (1); GA – множество процессов достижения целей; ga_i – элемент множества процессов достижения целей; g_j – цель; me_k – метаребро метаграфа, задающее процесс достижения цели.

Множество процессов достижения целей GA используется для связи целей системы с процессами, обеспечивающими их выполнение. Цели системы организованы в виде частично-упорядоченного множества. Метаребра метаграфа, задающие процессы достижения целей, позволяют определить иерархию процессов. Каждой цели ставится в соответствие процесс ее выполнения в виде метаребра метаграфа.

Описание метаграфового агента. Приведенный фрагмент прагматики информационной системы (см. рис. 4) показывает взаимодействие между элементами процесса и данными, но не дает информации о том, как должен быть представлен “активный” элемент процесса. В качестве активного элемента процесса в предлагаемой модели используется метаграфовый агент. Таким образом, каждый элемент процесса представляется или метаребром, которое содержит описание подпроцесса, или метавершиной, которая включает в себя описание метаграфового агента.

Основное назначение метаграфового агента — преобразование метаграфов. Особенностью метаграфового агента является то, что его описание также может быть представлено в форме метаграфа.

В соответствии с классификацией, приведенной в работе [12], метаграфовый агент относится к агентам реактивного типа и реализует поведение на основе заданного набора правил.

Определим метаграфовый $AG = \{ag_i\}$, $ag_i = \langle MG, R, AG^{ST} \rangle$, $R = \{r_j\}$, где AG — множество метаграфовых агентов; ag_i — метаграфовый агент; MG — метаграф, на основе которого выполняются правила агента; R — набор правил (множество правил r_j); AG^{ST} — стартовое условие выполнения агента (фрагмент метаграфа, который используется для стартовой проверки правил, или стартовое правило).

Структура правила метаграфового агента $r_i : MG_j \rightarrow OP^{MG}$, где r_i — правило; MG_j — фрагмент метаграфа, на основе которого выполняется правило; OP^{MG} — множество операций, выполняемых над метаграфом.

Антецедент правила — фрагмент метаграфа, консеквент правила — множество операций, выполняемых над метаграфом. Представление метаграфового агента в виде фрагмента метаграфа приведено на рис. 5.

Метаграфовый агент представлен в виде метавершины метаграфа. В соответствии с определением он связан с метаграфом MG_1 , на основе которого выполняются правила агента. Эта связь показана ребром e_4 .

Метаграфовый агент содержит множество вложенных метавершин, соответствующих правилам (правило 1 — правило N). Каждая метавершина правила содержит вершины антецедента и консеквента. В приведенном примере с антецедентом правила связана метавершина данных mv_2 (показано ребром e_2), а с консеквентом правила — метавершина данных mv_3 (показано ребром e_3). Условия срабатывания антецедента и множество действий консеквента задаются в виде атрибутов соответствующих вершин.

Стартовое условие выполнения агента задается с помощью атрибута “start = true”. Если стартовое условие задается в виде стартового правила, то данным атрибутом помечается метавершина соответствующего правила (в примере это правило 1). Если стартовое условие

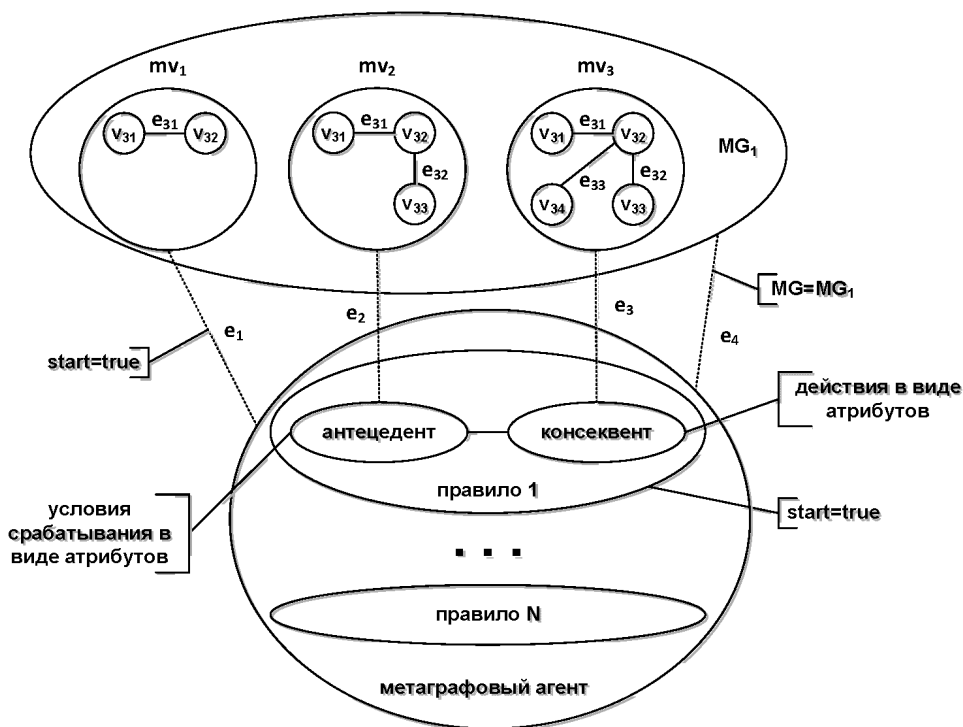


Рис. 5. Представление метаграфового агента в виде фрагмента метаграфа

записывается в виде стартового фрагмента метаграфа, который используется для стартовой проверки правил, то атрибутом “start = true” помечается ребро, которое связывает стартовый фрагмент метаграфа с метавершиной агента (в примере это ребро e_1).

Обсуждение полученных результатов. Использование метаграфов для представления семантики информационной системы позволяет избежать проблемы, присущих другим графовым моделям.

Модель на основе триплетов (RDF) не позволяет удобно описывать сложные объекты и N-местные отношения между вершинами графа. Для решения этой проблемы веб-консорциумом был предложен отдельный документ [9]. Использование метавершин позволяет моделировать как сложные объекты, так и N-местные отношения между вершинами графа.

Гиперграфы не целесообразно использовать для моделирования сложных иерархий между элементами, эта задача также удобно решается с помощью метавершин. Хотелось бы также отметить, что в некоторых работах термин “гиперграф” употребляется как синоним термина “метаграф”. В частности, в работе [10] автор в наименовании работы использует термин “гиперграф”, но в самой работе вводит понятие “семантический гиперграф”, структура которого больше напоминает структуру метаграфа.

Основное преимущество использования метаграфов для представления семантики информационной системы состоит в том, что с помощью метавершин можно легко аннотировать существующие вершины и связи дополнительными вершинами, не разрушая существующие семантические элементы и не перестраивая отношения между ними. Это позволяет в случае изменения прагматики информационной системы (при изменении целей информационной системы) с помощью метавершин гибко достраивать семантику информационной системы для реализации требуемых изменений.

В работе [11] предложено применение сетевой структуры (синергетической интеррепрезентативной сети) для моделирования сложных взаимодействий на основе событий, но не предложены подходы к моделированию данных. Подход на основе метаграфов позволяет моделировать прагматику информационной системы в виде единой структуры, содержащей данные, процессы и взаимосвязи между ними.

Идея описания процессов в виде метаребер метаграфа отчасти напоминает описание процессов в workflow-системе [13]. В настоящее время классические workflow-системы критикуют за жесткость связей, невозможность динамически адаптировать поведение workflow-системы. Для устранения недостатков классических workflow-систем предлагается подход, основанный на событиях.

В работе [14] такой подход называется событийно-ориентированной цепочкой процессов (event-driven process chain). В предлагаемом подходе одно входное событие влечет за собой выполнение одного процесса, который порождает одно выходное событие.

Более гибкий по сравнению с подходом, предложенным в работе [14], предлагается в работе [15] где событийно-ориентированные процессы рекомендовано строить с использованием системы производственных правил, которые реализуют проверку входных событий, выполнение действий и порождение выходных событий.

Подход на основе метаграфовых агентов более гибок, чем подходы, рассмотренные в работах [14, 15]. В нем не делается различий между специально генерируемыми событиями и изменениями данных, как те, так и другие представляются фрагментами метаграфа. Срабатывание правил метаграфового агента может быть обусловлено возникновением события в виде вершины метаграфа и изменением данных, которые также трактуются как фрагмент метаграфа. В отличие от workflow-подхода предусмотрена явная связь процессов с целями системы.

Новизна работы состоит в том, что авторами предложен подход к описанию семантики и прагматики информационных систем на основе метаграфов, предложена модель ИЭМ, введены основные операции над метаграфами в форме ИЭМ, предложен способ реализации прагматики с помощью метаграфового агента.

Заключение. Описание семантики и прагматики информационной системы — важные задачи при проектировании системы. В настоящее время для описания семантики используются графовые модели, однако существующие модели (такие как гиперграфы и RDF-триплеты) обладают ограничениями. Предлагаемый подход на основе метаграфов позволяет устранить эти ограничения. Для хранения и обработки метаграфа предложено использование ИЭМ.

Основные операции над метаграфами могут быть реализованы в виде API (программного интерфейса) в большинстве современных языков программирования. Предложен подход к описанию прагматики информационной системы в виде метаграфовой структуры, для описания данных используются метавершины, а для описания процессов — метаребра. Цели информационной системы представлены в виде частично-упорядоченного множества, связанного с метаграфовым описанием.

Активный элемент предложенной модели — метаграфовый агент, который относится к агентам реактивного типа и реализует поведение на основе заданного набора правил. К перспективам развития настоящей работы можно отнести более детальную проработку модели представления процессов на основе метаграфов.

ЛИТЕРАТУРА

1. Клыков Ю.И., Горьков Л.Н. Банки данных для принятия решений. М.: Сов. радио, 1980. 208 с.
2. Клыков Ю.И. Ситуационное управление большими системами. М.: Энергия, 1974. 136 с.
3. Поспелов Д.А. Большие системы. Ситуационное управление. М.: Знание, 1975. 64 с.
4. Поспелов Д.А. Ситуационное управление: теория и практика. М.: Наука, 1986. 288 с.
5. Мечковская Н.Б. Семиотика: Язык. Природа. Культура: Курс лекций. М.: Издательский центр “Академия”, 2008. 432 с.
6. Овчинников В.А. Графы в задачах анализа и синтеза структур сложных систем. М.: Изд-во МГТУ им. Н.Э. Баумана, 2014. 423 с.
7. Астанин С.В., Драгныш Н.В., Жуковская Н.К. Вложенные метаграфы как модели сложных объектов // Электронный научный журнал “Инженерный вестник Дона”. 2012. № 4. URL: <http://ivdon.ru/magazine/archive/n4p2y2012/1434> (дата обращения: 15.04.2014).
8. Basu A., Blanning R. *Metagraphs and Their Applications*. Springer, 2007. 174 p.
9. *Defining N-ary Relations on the Semantic Web*. W3C Working Group Note 12 April 2006. URL: <http://www.w3.org/TR/swbp-n-aryRelations/> (дата обращения: 15.04.2014).
10. Хахалин Г.К. Прикладная онтология на языке гиперграфов // Труды второй Всеросс. конф. с международным участием “Знания–Онтологии–Теории” (ЗОНТ–09). Новосибирск. 2009. С. 223–231.
11. Portugali J. The Seven Basic Propositions of SIRN // *Nonlinear Phenomena in Complex Systems*. 2002. Vol. 5:4. P. 428–444.

12. *Tarasov V.B.* От многоагентных систем к интеллектуальным организациям: философия, психология, информатика. М.: Эдиториал УРСС, 2002. 352 с.
13. *Биберштейн Н., Боуз С.* Компас в мире сервис-ориентированной архитектуры (SOA); пер. с англ. М.: Кудиц-Пресс, 2007. 256 с.
14. *Scheer A.-W., Thomas O., Adam O.* Process Modeling Using Event-Driven Process Chains. / М. Dumas, W.M.P. van der Aalst, A.H.M. ter Hofstede (Ed.) // Process-Aware Information Systems — Bridging People and Software through Process Technology. John Wiley & Sons, 2005. P. 119–146.
15. *An Event-Driven Workflow Engine for Service-based Business Systems* / J. Jang, A. Fekete, P. Greenfield, S. Nepal // Enterprise Distributed Object Computing Conference, 2006. EDOC'06. 10th IEEE International. P. 233–242.

REFERENCES

- [1] Klykov Yu.I., Gor'kov L.N. Banki dannykh dlya prinyatiya resheniy [Data banks for decision-making]. Moscow, Sov. Radio Publ., 1980. 208 p.
- [2] Klykov Yu.I. Situatsionnoe upravlenie bol'shimi sistemami [Situational control of large systems]. Moscow, Energiya Publ., 1974. 136 p.
- [3] Pospelov D.A. Bol'shie sistemy. Situatsionnoe upravlenie [Large systems. Situational control]. Moscow, Znanie Publ., 1975. 64 p.
- [4] Pospelov D.A. Situatsionnoe upravlenie: teoriya i praktika [Situational control: theory and practice]. Moscow, Nauka Publ., 1986. 288 p.
- [5] Mechkovskaya N.B. Semiotika: Yazyk. Priroda. Kul'tura [Semiotics: Language. Nature. Culture]. Moscow, Izdatel'skiy tsentr "Akademiya" Publ., 3th Ed., 2008. 432 p.
- [6] Ovchinnikov V.A. Grafy v zadachakh analiza i sinteza struktur slozhnykh system [Graphs in analysis and synthesis of structures for complex systems]. Moscow, MGTU im. N.E. Bauman Publ., 2014. 423 p.
- [7] Astanin S.V., Dragnysh N.V., Zhukovskaya N.K. Nested metagraphs as models of complex objects. *Elekt. Nauch. Zh. "Inzh. vestn. Dona"* [El. Sci. J. "Eng. J. of Don"], 2012, no. 4 (in Russ.). Available at: <http://ivdon.ru/magazine/archive/n4p2y2012/1434> (accessed 15.04.2014).
- [8] Basu A., Blanning R. Metagraphs and their applications. Springer, 2007. 174 p.
- [9] Defining *N*-ary relations on the semantic Web. *W3C Working Group Note*, 12 April 2006. Available at: <http://www.w3.org/TR/swbp-n-aryRelations/> (accessed 15.04.2014).
- [10] Khakhalin G.K. Applied ontology in hypergraphs language. *Tr. Vtoroy Vseross. Konf. "Znaniya-Ontologii-Teorii" (ZONT-09) s mezhdunarodnym uchastiem* [Proc. 2nd All-Rus. Conf. "Knowledge–Ontology–Theory" (KONT-09) with intern. participation], Novosibirsk, 2009, pp. 223–231 (in Russ.). Available at: <http://www.math.nsc.ru/conference/zont09/reports/21Hahalin.pdf> (accessed 15.04.2014).
- [11] Portugali J. The seven basic propositions of SIRN. *Nonlinear Phenomena in Complex Systems*, 2002, vol. 5, no. 4, pp. 428–444.
- [12] Tarasov V.B. Ot mnogoagentnykh sistem k intellektual'nym organizatsiyam: filosofiya, psikhologiya, informatika [From multi-agent systems to intelligent organizations: philosophy, psychology, computer science]. Moscow, Editorial URSS Publ., 2002. 352 p.
- [13] Bieberstein N., Bose S., Fiammante M., Jones K., Shah R. Service-Oriented Architecture (SOA) Compass: Business Value, Planning, and Enterprise Roadmap. IBM Press, 2005. 272 p. (Russ. ed.: Bibershteyn N., Bouz S., Fiammante M., Jones K., Shah R. Kompas v mire servis-orientirovannoy arkhitektury (SOA). Moscow, Kudits-Press Publ., 2007. 256 p.).

- [14] Scheer A.-W., Thomas O., Adam O. Process modeling using event-driven process chains. In book: *Process-aware information systems: bridging people and software through process technology*. Ed. by Dumas M., van der Aalst W.M.P., ter Hofstede A.H.M. John Wiley & Sons, 2005, 432 p. (pp. 119–146).
- [15] Jang J., Fekete A., Greenfield P., Nepal S. An event-driven workflow engine for service-based business systems. *Proc. Enterprise Distributed Object Computing Conf., 2006. EDOC '06. 10th IEEE Int.*, Hong Kong, China, 2006, pp. 233–242.

Статья поступила в редакцию 22.04.2014

Самохвалов Эдуард Николаевич — канд. техн. наук, профессор кафедры “Системы обработки информации и управления” МГТУ им. Н.Э. Баумана. Автор более 50 научных работ в области проектирования информационных систем в области обучения. МГТУ им. Н.Э. Баумана, Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5.

Samokhvalov E.N. — Cand. Sci. (Eng.), professor of “Information Processing System and Control” department of the Bauman Moscow State Technical University. Author of more than 50 publications in the field of designing of E-Learning Automated Systems. Bauman Moscow State Technical University, 2-ya Baumanskaya ul. 5, Moscow, 105005 Russian Federation.

Ревунков Георгий Иванович — канд. техн. наук, доцент кафедры “Системы обработки информации и управления” МГТУ им. Н.Э. Баумана. Автор более 40 научных работ в области баз данных, проектирования автоматизированных систем. МГТУ им. Н.Э. Баумана, Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5.

Revunkov G.I. — Cand. Sci. (Eng.), assoc. professor of “Information Processing System and Control” department of the Bauman Moscow State Technical University. Author of more than 40 publications in the field of databases, designing of automated systems. Bauman Moscow State Technical University, 2-ya Baumanskaya ul. 5, Moscow, 105005 Russian Federation.

Гапанюк Юрий Евгеньевич — канд. техн. наук, доцент кафедры “Системы обработки информации и управления” МГТУ им. Н.Э. Баумана. Автор более 15 научных работ в области проектирования автоматизированных систем. МГТУ им. Н.Э. Баумана, Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5.

Gapanyuk Yu.E. — Cand. Sci. (Eng.), assoc. professor of “Information Processing System and Control” department of the Bauman Moscow State Technical University. Author of more than 15 publications in the field of designing of automated systems. Bauman Moscow State Technical University, 2-ya Baumanskaya ul. 5, Moscow, 105005 Russian Federation.