

МАТЕМАТИЧЕСКИЕ МОДЕЛИ КОНФИГУРИРОВАНИЯ СЕТЕЙ ИСТОЧНИКОВ ЗНАНИЙ В БИЗНЕС-СРЕДЕ

Н. Г. Шилов

Санкт-Петербургский институт информатики и автоматизации РАН
199178, Санкт-Петербург, 14-я линия В.О., д.39

nick@mail.iias.spb.su

УДК 681.3

Н. Г. Шилов. Математические модели конфигурирования сетей источников знаний в бизнес-среде // Труды СПИИРАН. Вып. 1, т. 3. — СПб: СПИИРАН, 2003.

Аннотация. Настоящая статья предлагает математические модели, позволяющие оптимально использовать источники знаний, на основе заданных критериев — Библ. 10 назв.

UDC 681.3

N. Chilov. Mathematical Models for knowledge source network configuration in business environment // SPIIRAS Proceedings. Issue 1, v. 3. — SPb: SPIIRAS, 2003.

Abstract. The paper presented suggests mathematical models for optimal utilizing of knowledge sources according to criteria specified. — Bibl. 10 items.

1. Введение

В настоящее время непрерывно растущие темпы рыночных отношений, вызвавшие появление таких форм кооперации, как виртуальные предприятия, заставляют организации эффективнее и быстрее использовать и модифицировать имеющиеся ресурсы, как материальные (производственные запасы, производственные мощности, и т.д.), так и нематериальные (технологии, информация, знания). Усиливающаяся конкуренция, как на внутренних, так и внешних рынках, также требует повышения мобильности и гибкости.

С целью информационной поддержки процессов принятия решений в подобных направлениях разрабатываются системы управления ресурсами предприятия (ERP Systems), которые объединяют управление как материальными (например, материальные запасы и оборудование), так и нематериальными ресурсами (например, знания и навыки). В качестве примера такой системы можно назвать SAP [1]. Однако, они все еще не могут обеспечить оптимального конфигурирования производственных систем и их подсистем.

В данной статье рассматриваются вопросы, связанные с управлением таким видом нематериальных ресурсов, как знания, который является весьма важным для выполнения требований, упомянутых выше. С целью определения понятия «знание», можно выделить три его основных типа [2]:

- 1) описательное знание (Descriptive Knowledge) — описание прошлого, настоящего или некоторого гипотетического состояния рассматриваемой области интересов;
- 2) процедурное знание (Procedural Knowledge) — описание технологий, методов, алгоритмов, т.е. того, как выполнять тот или иной процесс;
- 3) аргументирующее знание (Reasoning Knowledge) — описание границ, в которых определенные заключения являются верными в определенных условиях.

Однако, поскольку для работы со знаниями необходимо задать метод их описания, в дальнейшем, исходя из объектного представления, они будут определены, как объекты, методы и отношения, предположительно существую-

щие в некоторой проблемной области. Причем под проблемной областью будет пониматься бизнес-среда, т.е. субъекты рынка, их поведение, отношения между ними, а также их конфигурирование.

2. Постановка задачи

В общем виде задача оптимизации какого-либо ресурса может быть описана следующим образом: необходимо определить эффективное размещение в пространстве (т.е., где находится) и времени (т.е., когда есть в наличии, когда доступен) составляющих данного ресурса, а также когда и какую составляющую использовать. В качестве критерия могут выступать такие показатели, как стоимость получения и доставки ресурсов и/или время, необходимое для этого. Кроме того, необходимо учитывать взаимосвязи между ресурсами (например, ресурс может состоять из других ресурсов), наличие которых позволяет говорить о данной задаче, как о сети. Иллюстрация сетевой постановки такой задачи изображена на рис. 1, где вершины сети — ресурсы, а дуги — связи между ними (перемещение, использование и т.п.).

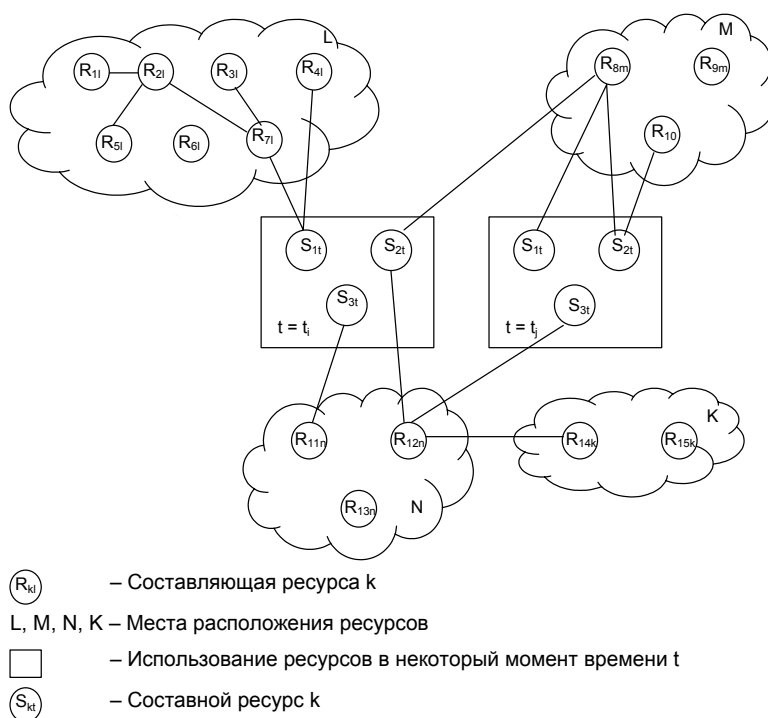


Рис. 1. Иллюстрация концептуальной модели сети ресурсов в сетевой постановке

Под понятием составного ресурса подразумевается изначально несуществующий ресурс, для формирования которого используются другие ресурсы. Например, производственные мощности создаваемого виртуального предприятия, которые формируются из производственных мощностей его участников или изначально несуществующий информационный ресурс, который можно создать с использованием других информационных ресурсов.

Применительно к знаниям в качестве составляющих ресурса будем рассматривать источники знания. Выделяют два типа источников знаний [2]:

- 1) Схематические источники знания (Schematic Knowledge Resource) — инфраструктура, культура, стратегия и цель существования организации;
- 2) Содержательные источники знания (Content Knowledge Resource) — знание участников организации, а также знания, хранимые на других носителях (например, компьютерные базы знаний).

В дальнейшем под источниками знаний будем понимать только содержательные источники.

Итак, задачей является построение сети источников знаний эффективной, согласно некоторым заданным критериям и допустимой по наложенному набору ограничений структуры, а также определение правил нахождения того, какой из источников, как и когда следует использовать. Для этой цели необходим аппарат, включающий модели и алгоритмы, позволяющие находить решения для такого рода задач.

3. Сети источников знаний

Сеть ресурсов, в узлах которой находятся источники знаний, будем называть сетью источников знаний (рис. 2) [3], [4]. Компоненты, непосредственно используемые для формирования ресурса, будем называть компонентами первого уровня (R_{1i} , R_{6m} и R_{10n}), а источники, их содержащие, — источниками первого уровня (L , M и N). Характерные особенности данной сети следующие:

- 1) Сеть является распределенной. Ее узлы (источники знаний) могут быть территориально удалены друг от друга. Информация об источниках и их расположении хранится в картограмме знаний.
- 2) Сеть является динамической. Набор узлов и их характеристик изменяется с течением времени, и, следовательно, картограмма знаний также является динамической.
- 3) Узлы сети имеют различную природу. Данное свойство следует из определения содержательных источников знания. Например, можно выделить следующие:
 - базы знаний;
 - эксперты;
 - базы данных;
 - структурированные документы (например, текстовые и HTML документы);
 - прочие источники, для которых используются некоторые механизмы распознавания информации и извлечения знаний.

Задачу конфигурирования такой сети можно декомпозировать следующим образом. При возникновении потребности в том или ином знании на узлах с помощью данных картограммы знаний строится сеть (*выбор источников знаний для удовлетворения запроса пользователя*). Затем определяется, в какие моменты времени будут получены ответы от источников, и планируется их взаимодействие (*координация сети источников знаний*). Кроме того, можно добавить следующие две подзадачи: определение того, какое знание и в каком количестве хранить в каждом из источников (*распределение знаний в сети источников*), а также *определение объема источников знаний*. Далее будут рассмотрены первые две из перечисленных задач.

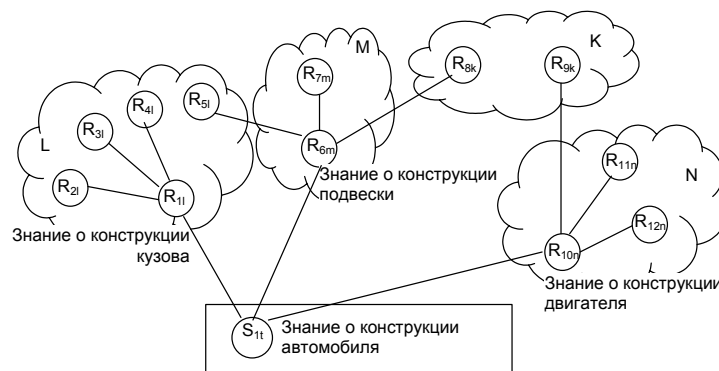


Рис. 2. Пример сети источников знаний

4. Выбор источников знаний для удовлетворения запроса пользователя

Задача выбора источников знаний для удовлетворения запроса пользователя заключается в том, чтобы выбрать те источники знаний, которые будут использоваться для наиболее эффективного формирования ответа на запрос пользователя. Существует ряд проектов, посвященных решению задач, связанных с поиском знаний для удовлетворения запроса, например, KRAFT [5], InfoSleuth [6], OBSERVER [7]. Однако, они предоставляют весьма ограниченные механизмы, ориентированные на поиск наиболее эффективных решений, отвечающих заданным критериям (например, по стоимости или времени), или оценки эффективности найденного решения. В качестве возможного подхода к решению этой задачи предлагается математическая модель, описанная ниже.

Изначально в системе имеются:

Онтология проблемной области (A), содержащая некоторые элементы знания (r_j), т.е. объекты (O), атрибуты (V), домены (D) и ограничения (C) проблемной области. В данной постановке задачи будем рассматривать только элементы первого уровня, т.е. без учета связей между элементами знаний.

$$A = (O, V, D, C) = \{r_j\}, \quad j = 1 \dots n, \quad \text{где} \quad (4.1)$$

n — количество элементов знания в антологии проблемной области.

Источники знаний (S_i), содержащие некоторые элементы знаний о проблемной области (K_{jit}), на некоторый момент времени (t). Кроме элементов знаний, включенных в онтологию, в источниках знаний содержатся их экземпляры (instances), т.е. некоторая информационно-содержательная составляющая (I)

$$S_{it} = (O, V, D, C, I) = \{R_{jit}\}, \quad i = 1 \dots m, \quad t = 1 \dots T, \quad \text{где} \quad (4.2)$$

m — количество источников знаний в системе;

T — время функционирования системы.

Картограмма знаний, ставящая элементы знаний из источников в соответствие с понятиями онтологии проблемной области на некоторый момент времени (t). Соответствие будем обозначать символом « \rightarrow », а утверждение «элементу знания r_j соответствует элемент знания R_{jit} » парой ($r_j \rightarrow R_{jit}$):

$$KM_t = \{(r_j \rightarrow R_{jit})\}, \quad \text{где } r_j \in A \text{ и } R_{jit} \in S_{it} \quad (4.3)$$

Считается, что для каждого источника известны такие его параметры, как стоимость извлечения знаний, расписание работы и т.д. Кроме того, определим онтологию данного источника, как соответствие элементов знаний источника i понятиям онтологии проблемной области в рамках данного источника:

$$A(S_{it}) = \{(r_j \rightarrow R_{jit})\} \quad (4.4)$$

При попадании запроса (Q) в систему, он декомпозируется на некоторое множество подзапросов (q_k), которые затем переводятся в понятия системы, т.е. в понятия онтологии проблемной области. Соответствие подзапросов понятиям системы назовем онтологией запроса ($A(Q)$). После выполнения данных операций будет получен запрос (Q'), описанный в понятиях системы, т.е. запрос, разбитый на подзапросы, соответствующие понятиям системы:

$$Q = \{q_k\} \quad (4.5)$$

$$A(Q) = \{(q_k \rightarrow r_j)\}, \text{ где } q_k \in Q \text{ и } r_j \in A \quad (4.6)$$

$$Q' = \{r_j\}, \text{ где } r_j \text{ такие, что } \exists (q_k \rightarrow r_j) \in A(Q) \quad (4.7)$$

После вышеуказанных преобразований допустимое решение задачи выбора источников знаний (Dec_Q) первого уровня, т.е. без учета источников, с которыми они связаны, может быть описано следующим образом:

$$Dec_Q = \{(r_j \rightarrow R_{jit})\}, \text{ где } r_j \in Q', \text{ или} \quad (4.8)$$

$$Dec_Q = \{(q_k \rightarrow R_{jit})\} \quad (4.9)$$

В качестве оценки оптимальности решения можно рассмотреть такие показатели, как затраты ($Cost$), связанные с выполнением запроса, и время ($Time$) на его обработку:

$$Cost = f_{Cost}(Dec_Q) = \sum_{R_{jit} \in Dec_Q} f_{Cost}(R_{jit}) \quad (4.10)$$

$$Time = f_{Time}(Dec_Q) \quad (4.11)$$

Кроме того, построить обобщенный показатель эффективности (Eff) решения, включающий оценки как затрат, так и времени (многокритериальная оптимизация):

$$Eff = f_{Eff}(Dec_Q) = f'_{Eff}(f_{Cost}(Dec_Q), f_{Time}(Dec_Q)) \quad (4.12)$$

Например, можно получить нормализованные значения функций затрат и времени (с индексом N), и, построить их аддитивную свертку с использованием весов w_{Cost} и w_{Time} :

$$\begin{cases} Eff = f_{Eff}(Dec_Q) = f'_{Eff}(f_{Cost}(Dec_Q), f_{Time}(Dec_Q)) = \\ = w_{Cost} \cdot f_{Cost}^N(Dec_Q) + w_{Time} \cdot f_{Time}^N(Dec_Q) \\ w_{Cost} + w_{Time} = 1 \end{cases} \quad (4.13)$$

Таким образом, эффективным решением (Dec_Q^{eff}) будет называться решение, при котором значение выбранной целевой функции (например, многокритериальной) будет минимальным, с учетом выполнения структурных ограничений, описанных выше:

$$Dec_Q = Dec_Q^{eff}, \text{ если } f_{Eff}(Dec_Q) \rightarrow \min \quad (4.14)$$

При включении в модель источников более удаленных («глубоких») уровней, т.е. источников опосредованно связанных с запросом через другие источники, модель необходимо расширить, построив и добавив аналогичные модели, где в качестве запроса будут выступать источники знаний первого уровня.

5. Координация сети источников знаний

Задача координации сети источников знаний заключается в нахождении наиболее эффективного плана того, в какие моменты времени должны быть посланы запросы источников знаний друг другу, сгенерированы ими и переданы ответы в процессе нахождения или построения ответа на запрос пользователя, при заданных структурных ограничениях, описанных в предыдущем разделе.

Пусть наличие/отсутствие ресурса R_j в источнике i в момент времени t обозначается переменной RA_{jit} . Новое знание в начале процесса его получения (генерации) — RG_{jit} . Знание в начальный момент процесса его передачи одним источником (k) другому (i) — RT_{jikt} . Все вышеупомянутые переменные могут принимать значения 1 или 0. Тогда уравнение (5.1) будет определять условия доступности ресурса R_j для источника знаний i . Данное условие заключается в следующем: ресурс является доступным для источника знаний, если выполнено хотя бы одно из следующих условий: (i) ресурс уже содержится в данном источнике, (ii) ресурс сгенерирован данным источником, (iii) ресурс был получен данным источником из другого источника j .

$$RA_{j,i,t} = RA_{j,i,t-1} \vee RG_{j,i,t+TG_{ij}} \vee RT_{j,k,i,t+TT_{kj}}, \quad k = 1 \dots m, \quad \text{где} \quad (5.1)$$

TG_{ij} — время, необходимое для генерации нового знания j в источнике i . Поскольку оценка этого параметра достаточно сложна и субъективна, для этого предлагается применение значений, полученных путем экспертных оценок с использованием системы групповой поддержки принятия решений [8].

TT_{ikj} — время, необходимое для передачи знания j из источника k в источник i .

Выражение (5.2) описывает возможность генерации ресурса на основе имеющихся ресурсов, а именно: ресурс может быть сгенерирован, только при условии доступности всех ресурсов, требуемых для баланса:

$$RG_{jit} \leq (RA_{1it} \vee \neg RR_{1ji}) \wedge (RA_{2it} \vee \neg RR_{2ji}) \wedge \dots \wedge (RA_{nit} \vee \neg RR_{nji}), \quad \text{где} \quad (5.2)$$

RR_{lji} — переменная, принимающая значение 1, если ресурс l требуется для генерации знания j источником i , и значение 0 в противном случае.

Также необходимо учесть ограничения на возможности источников по хранению (5.3) и генерированию (5.4) знаний (приведенные формулы относятся к источнику i и периоду времени t), а также на пропускную способность средств передачи знаний между источниками (5.5), получения знаний (5.6) и передачи знаний (5.7). Формулы приведены для источников i и k . Для оценки этих параметров также предлагается использовать экспертные оценки.

$$\sum_{i=l} RA_{jit} \leq RA_{\max lt} \quad (5.3)$$

$$\sum_{i=l} RG_{jit} \leq RG_{\max lt} \quad (5.4)$$

$$\sum_{i=l, r=k} RT_{jirt} \leq RT_{\max lkt} \quad (5.5)$$

$$\sum_{i=l} RT_{jrit} \leq RTI_{\max lt} \quad (5.6)$$

$$\sum_{i=l} RT_{jirt} \leq RTO_{\max lt} \quad (5.7)$$

Целевая функция может быть, как и в предыдущем случае одного из трех типов: минимизация затрат (5.8), минимизация времени (5.9) и многокритериальная минимизация затрат и времени (5.10)

$$C = f_C(D_Q) = \sum_t \sum_i \left[CRA_{it} \cdot \sum_j RA_{ijt} + CRTI_{it} \cdot \sum_r \sum_j RT_{jrit} + CRTO_{it} \cdot \sum_r \sum_j RT_{jirt} + \sum_r CRT_{irt} \cdot \left(\sum_j RT_{jirt} \right) \right], \quad (5.8)$$

где

CRA_{it} — затраты на содержание единицы ресурса источником i в момент времени t ;

$CRTI_{it}$ — затраты на отправку единицы ресурса источником i в момент времени t ;

$CRTO_{it}$ — затраты на получение единицы ресурса источником i в момент времени t ;

CRT_{ikt} — затраты на передачу единицы ресурса между источниками i и r в момент времени t .

$$T = f_T(D_Q) = \max_{R_j \in D_Q} (RA_{ijt}), \quad \text{где} \quad (5.9)$$

i — источник, предоставляющий ресурсы конечному потребителю.

$$E = f_E(D_Q) = f'_E(f_C(D_Q), f_T(D_Q)) \quad (5.10)$$

6. ILOG Configurator, как средство реализации предложенных моделей

Для реализации вышеописанных моделей предлагается использование средств пакета ILOG Configurator [9]. Данный пакет представляет собой набор библиотек для C++ и Java, содержащих объекты и методы для постановки и решения задач конфигурирования (в том числе конфигурирования сетей ограничений). Для выполнения оптимизации ILOG Configurator использует средства другого пакета компании ILOG – ILOG Solver. Solver также является набором библиотек для C++ и Java, но ориентирован на решение задач оптимизационного характера. Основным его назначением является решение нелинейных задач методами направленного перебора.

В ILOG Configurator задача представляется в объектном виде. Система поддерживает следующие два типа отношений (рис. 3): «является экземпляром» («is a») между типами (types) и объектами (instance), и «связан с» («is connected to») между объектами. В свою очередь, отношения «связан» могут быть видов «часть-целое» («has part» или «part of») и «использует» («uses»). Ограничения типа «связан с», фактически устанавливаются между классами, содержащими объекты, в виде переменных, которые принимают значения, связываемых объектов.

Также ILOG Configurator поддерживает иерархическую структуру типов, в которой общие типы содержат более специализированные. Пример, изображенный на рис. 4 содержит иерархию типов, состоящую из 3-х уровней.

В данном примере тип «Форд Фокус» содержит подтипы «Форд Фокус седан» и «Форд Фокус универсал», которые содержат более узкие подтипы – модели автомобилей седан LX, седан SE, универсал SE и универсал Street Edition SE. Объекты или конкретные автомобили (не представлены на рисунке) являются экземплярами типов низшего уровня (в данном случае – моделей автомо-

билей). Отношения между объектами и типами среднего уровня, а также отношения между типами среднего уровня и типом верхнего уровня являются отношениями типа «является». Например, модель LX является автомобилем типа Форд Фокус седан, который, в свою очередь, является автомобилем типа Форд Фокус. Отношения типа «связан с» можно проиллюстрировать на следующих примерах. Отношением «часть-целое» (иерархическое отношение) связаны автомобиль Форд Фокус LX и используемый в нем двигатель 2.0L SPI. Отношением «использует» (отношение одного уровня) связаны двигатель и кузов автомобиля (они не являются частями друг друга, но имеют ряд параметров, влияющих на их совместимость, например расположение узлов крепления).

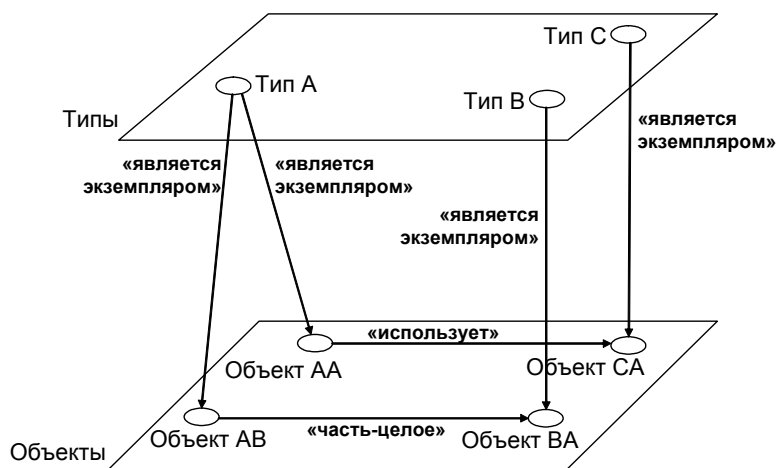


Рис. 3. Иллюстрация отношений в системе ILOG Configurator

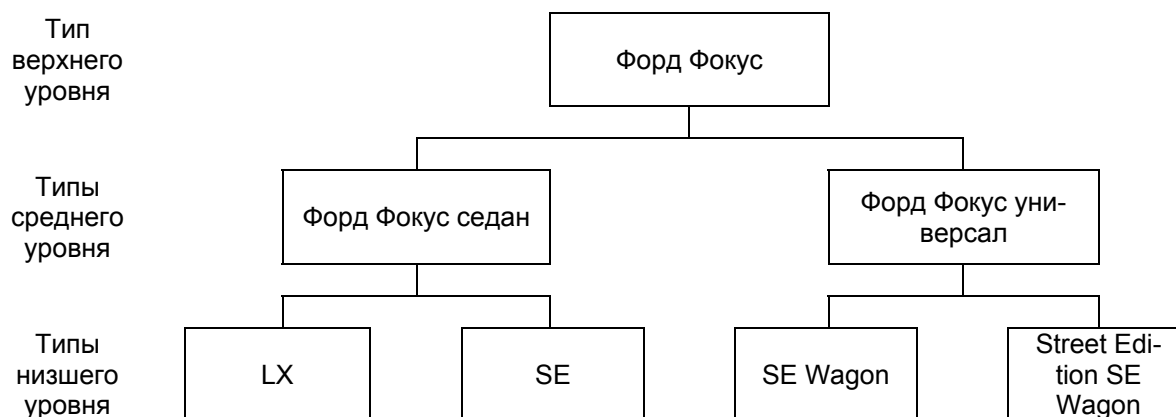


Рис. 4. Пример иерархии классов в системе ILOG Configurator

ILOG Configurator позволяет задавать атрибуты для типов и присваивать им значения для объектов. На эти значения можно накладывать ограничения 3-х типов:

- 1) Ограничения на целочисленные или вещественные значения переменных или использующих их выражений (арифметические, символические, логические, табличные);
- 2) Ограничения на множества переменных (объединение, пересечение, подмножество, «элемент содержится во множестве», «элемент не содержится во множестве»);

3) Метаограничения, или «ограничения на ограничения», позволяющие устанавливать веса значимости для других ограничений, тем самым, позволяя реализовать элементы экспертной оценки.

Кроме вышеупомянутых ограничений, ILOG Configurator позволяет использовать ограничения на совместимость типов. Данная возможность реализована с помощью так называемых таблиц совместимости и несовместимости типов, работающие по одному из следующих принципов:

- Кортеж, записанный в «таблице несовместимости», считается неверным подбором компонент. Только записанные кортежи считаются неверными, все остальные комбинации считаются верными;
- Кортеж, записанный в «таблице совместимости», считается верным подбором компонент. Только записанные кортежи считаются верными, все остальные комбинации считаются неверными.

Также, таблицы способны учитывать иерархию типов. Можно записывать кортежи на типы средних уровней, создавая следующие таблицы типов:

- Когда тип T является несовместимым с типом W, все дочерние типы T будут несовместимы с дочерними типами W;
- Когда тип T является совместимым с типом W, все дочерние типы T будут совместимы с дочерними типами W.

Ниже представлена иллюстрация данного ограничения. Табл. 1 является таблицей соответствия моделей автомобилей и двигателей, которые могут быть на них установлены. Ниже приведены примеры совместимых и несовместимых типов.

Таблица 1. Описание совместимости моделей и двигателей

Модель	Двигатель
Совместимые	
Focus Sedan LX	2.0L SPI
Focus Sedan LX	2.0L Zetec
Focus Sedan SE	2.0L SPI
Focus Sedan SE	2.0L Zetec
Focus Wagon SE	2.0L Zetec
Focus Wagon Street Edition	2.0L Zetec

Например, модель Focus Wagon SE совместима с двигателем (может иметь двигатель) 2.0L Zetec, но не совместима с двигателем (не может иметь двигатель) 2.0L SPI.

Заключение

Использование предложенных математических моделей поиска знаний и координирования сетей знаний позволит эффективно использовать источники знаний, исходя из заданных критериев и с учетом накладываемых ограничений. Последующие доопределения и усовершенствование моделей позволят более детально описать приведенные операции на языке онтологий [10], а их представление в виде сетей ограничений обеспечит возможность внедрения с использованием средств ILOG Configurator. Кроме того, в будущем планируется описание и реализация остальных подзадач поставленной проблемы.

Литература

- [1] Интернет сайт SAP AG. 2001. Интернет адрес <http://www.sap.com>.
- [2] *Holsapple C. W.*, The knowledge chain model: activities for competitiveness. – Expert Systems with Applications. — v. 20 — Стр. 77-98.
- [3] *Smirnov A. V.*, Rapid Knowledge Fusion into the Scalable Infosphere: A Concept and Possible Manufacturing Applications. — Proceedings of the International NAISO Congress on Information Science Innovations (ISI'2001), Symposium on Intelligent Automated Manufacturing (IAM'2001), Dubai, U.A.E., 2001.
- [4] *Смирнов А. В., Пашкин М. П., Левашова Т. В., Шилов Н. Г.*, Основные принципы организации систем быстрой интеграции знаний. — Труды СПИИРАН. Вып. 1, т. 2 — СПб: СПИИРАН, 2002.
- [5] *Preece F., Hui K., Gray A., Marti P., Bench-Capon T., Jones D., Cui Z.*, The KRAFT architecture for knowledge fusion and transformation. — Knowledge-Based Systems, 13, 2000. — Стр. 113–120
- [6] *Woelk D., Tomlinson C.*, The InfoSleuth Project: Intelligent Search Management via Semantic Agents. — Electronic Proceedings of the Second World Wide Web Conference '94: Mosaic and the Web, 1994.
- [7] *Mena E., Kashyap V., Sheth A., Illarramendi A.*, OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies Distributed and Parallel Databases. — V. 8, № 2, 2000. — Стр. 223-271.
- [8] *Пашкин М. П.*, Ранжирование альтернативных источников знаний на основе технологии групповой поддержки принятия решений. — Труды СПИИРАН. Вып. 1, т. 3 — СПб: СПИИРАН, 2003.
- [9] ILOG Configurator. Whitepaper. 2001. Интернет адрес http://www.ilog.com/products/configurator/wp_configurator.pdf.
- [10] *Левашова Т. В.*, Принципы управления онтологиями, используемые в среде интеграции знаний. — Труды СПИИРАН. Вып. 1, т. 2 — СПб: СПИИРАН, 2002.