

© 2012 г. А.Г. ОЛЕЙНИК, д-р техн. наук  
А.С. НЕВЕДРОВ

(Институт информатики и математического моделирования технологических процессов КНЦ РАН, Апатиты)

## АВТОМАТИЗАЦИЯ ПОИСКА АЛЬТЕРНАТИВНЫХ СПЕЦИФИКАЦИЙ ИСПОЛНИТЕЛЬНОЙ СРЕДЫ<sup>1</sup>

В данной работе рассматривается задача определения эффективных режимов обогащения минеральных руд. Представлен подход организации поиска потенциальных инструментальных средств на основе прикладной онтологии. Описан способ представления спецификаций исполнительной среды в виде графа инструментальных средств для поиска альтернатив.

**AUTOMATION SEARCHING OF ALTERNATIVE SIMULATION ENVIRONMENT SPECIFICATIONS** / A.G. Oleynik, A.S. Nevedrov (Institute for Informatics and Mathematical Modelling of Technological Processes of the Kola Science Center, Fersmana 24A, Apatity 184209, Russia, E-mail: nevedrov@arcticsu.ru). In the paper task of searching effective regime of ore separation is considered. The approach of searching potential toolkits based on applied ontology is proposed. The method of representation of simulation environment specifications as a graph of toolkits for searching of alternatives is described.

### 1. Введение

Для получения товарной продукции требуемого качества горно-обогатительные предприятия постоянно стремятся совершенствовать технику и технологии добычи и переработки рудного сырья [1]. При исследовании и разработке новых технологий обогащения в настоящее время активно используется компьютерное моделирование. Это позволяет сократить общее время исследования и материальные затраты на поиск и сравнительный анализ новых решений. Описание общей структуры и основных этапов решения задачи синтеза схем эффективных процессов разделения минерального сырья представлено в работе [2].

Многие процессы обогащения твердых полезных ископаемых основаны на различном воздействии в рабочем объеме разделительного аппарата сепарационных сил на разнородные частицы руды, находящиеся в жидкой среде. Современный уровень развития информационных технологий, в частности – создание САЕ-систем (Computer-Aided Engineering), позволяет использовать для исследования таких процессов компьютерные модели гидродинамических течений [3]. В этих моделях исходные дифференциальные уравнения в частных производных, описывающие физические процессы, заменяются соответствующими системами конечно-разностных уравнений

---

<sup>1</sup> Работа выполнена при финансовой поддержке РФФИ (проект № 12-07-98800-р\_север\_a).

большой размерности, решение которых требует большого объема вычислений. В связи с этим, для получения результатов задачи моделирования в приемлемое время при наличии ограничений на доступные вычислительные ресурсы необходимо найти наиболее рациональные варианты организации вычислительного процесса. Обобщенная схема вычислительного эксперимента, основанного на использовании метода конечных элементов для описания разделительного процесса, включает пять основных этапов [4]: создание геометрии моделируемого пространства; наложение расчетной сетки; задание начальных параметров и условий моделируемого процесса; собственно вычисления; представление результатов.

Одним из механизмов, обеспечивающих ускорение вычислений, является распараллеливание расчетов. Возможность применения этого механизма обусловлена структурой модели гидродинамических течений в объеме разделительного аппарата. Второе направление рационализации вычислительного эксперимента связано с формированием эффективной структуры исполнителей – программ и аппаратных средств, которые можно использовать в каждом конкретном случае.

Для решения поставленной проблемы была предложена концепция и архитектура информационной системы, которая должна обеспечить интегрированное представление разнородных исполнителей и систематизацию классов решаемых задач. Формализация описаний задач и ресурсов позволяет реализовать автоматизированный синтез спецификации исполнительской среды моделирования при существующих ограничениях [5]. Одним из важнейших и перспективных направлений в области формализации знаний, которое дает возможность использования накопленных знаний для компьютерной обработки, являются онтологии.

В настоящей работе в качестве формальной основы для автоматизированного синтеза спецификаций исполнителей предлагается использовать прикладную онтологию, представляющую систему понятий решаемых задач. Онтология позволяет произвести декомпозицию сложной задачи до подзадач, реализация которых возможна с помощью существующих программных средств. Анализ решения исходной задачи, реализуемый процедурами вывода на онтологии, дает возможность в автоматизированном режиме определить множество необходимых исполнителей и связей между ними, т.е. сформировать спецификацию исполнительской среды. Генерируемая спецификация в некоторых случаях может обладать избыточностью за счет альтернативных путей решения задачи, обусловленных наличием аналогов среди исполнителей. В итоге, решение исходной задачи можно произвести по одному из альтернативных путей, выбор которого определяется требованиями конкретного вычислительного эксперимента (время выполнения, точность решения и др.).

## **2. Структура прикладной онтологии**

Сегодня в теории принято классифицировать онтологии по степени зависимости от задач или прикладной области, по языку представления онтологических знаний и его выразительным возможностям и другим параметрам [6]. Прикладные онтологии описывают концепты, которые зависят как от онтологии задач, так и от онтологии предметной области. Онтологический инжиниринг при этом подразумевает: определение классов понятий в онтологии; определения таксономии на классах; разработку структур понятий и ситуаций; определение свойств понятий и значений этих свойств; процедуры вывода и преобразования ситуаций. Концепты в таких онтологиях часто соответствуют ролям, которые играют объекты в предметной области в процессе выполнения определенной деятельности.

Основываясь на идеях, изложенных в [7, 8], под онтологией мы будем понимать однозначное формальное описание (модель) некоторой части мира применительно к конкретной области интересов. Дадим определения основных понятий используемых в работе.

*Определение 1. Онтология – формальная спецификация объектов и связей между ними.*

*Определение 2. Объект – формальная структура сущности предметной области.*

*Определение 3. Свойство объекта – характеристика той или иной особенности объекта.*

*Определение 4. Экземпляр объекта – объект с непустыми значениями свойств, связанный с другим объектом отношением “быть экземпляром”.*

Свойство объекта, согласно определению 3, представим как совокупность следующих характеристик объекта:

$$(1) \quad p = \langle n_o, t_o, v_o \rangle$$

где

$n_o$  – символьное имя свойства  $p$  объекта  $o$ ,

$t_o$  – тип значения,

$v_o$  – значение свойства.

Объект, согласно определению 2, представим как следующую совокупность характеристик моделируемой сущности предметной области:

$$(2) \quad o = \langle N, P, S, D, I \rangle$$

где

$N$  – имя объекта,

$P$  – множество свойств объекта,

$S$  – суперклассы объекта,

$D$  – подклассы объекта,

$I$  – экземпляры объекта.

Множество базовых отношений между объектами онтологии представим с помощью следующего выражения:

$$(3) \quad R = \langle subclass, superclass, instance, usein, useout, model \rangle$$

где

*subclass* – отношение «быть подклассом»,

*superclass* – отношение «быть суперклассом»,

*instance* – отношение «быть экземпляром класса»,

*usein* – отношение «используется на входе»,

*useout* – отношение «используется на выходе»,

*model* – отношение «моделируется (реализуется) в».

С учетом вышесказанного, онтология представляется как совокупность структурно-логических схем свойств и объектов, связанных между собой отношениями  $R$ :

$$(4) \quad \text{Ontology} = \langle O, P_{ont}, R \rangle$$

где

$O$  – множество объектов, связанных отношениями из  $R$ ,

$P_{ont}$  – множество свойств онтологии.

В контексте данной работы прикладная онтология представляется в виде взаимосвязанных классов: сепарационные аппараты; процессы, протекающие в них; руды (сырье); инструментальные средства, моделирующие процессы. Кроме этого в явном виде определяется класс «Задача», содержащий спецификацию решаемой задачи. Каждый класс описывает систему понятий одного раздела предметной области (рисунок 1).



Рис. 1. Структура прикладной онтологии

В классе «Сепарационные аппараты» содержится описание аппаратов, как существующих, так и «виртуальных» - для которых существуют только компьютерные модели. Добавление в онтологию экземпляра данного класса в виде новой геометрии при решении задачи позволяет упростить решение последующих задач, объединить наработки различных исследовательских групп и т.д. Для классификации геометрий сепарационных аппаратов по сложности учитываются элементы (2d- или 3d-формы), из которых она состоит. Данная сложность будет учитываться при установлении отношений «моделируется в» с объектами подклассов класса «Инструментальные средства». Также, основываясь на типах аппаратов, устанавливается связь с процессами, которые могут протекать в них. Это в свою очередь определяет множество потенциальных исполнителей, способных моделировать процесс в сепарационном аппарате.

Класс «Процессы» включает в себя концепты, связанные с физическими и химическими процессами, проходящими в сепараторе во время разделения минеральных руд. Процессы характеризуются силами, действующими в аппарате, - сепарационными и так называемыми «фоновыми» (постоянно присутствующие силы, например, гравитационная сила). На данный момент существуют разработки онтологий по химии и

физике [9, 10], импорт которых возможен в данный класс при некоторых условиях. Объекты класса «Процессы» связаны с объектами класса «Решатели» отношением «моделируется в».

Класс «Руды» содержит описание минеральных руд и продуктов разделения – их характеристики (минеральный состав, крупность, плотность и другие сепарационные свойства). Объекты этого класса также связаны с объектами класса «Решатели» отношением «моделируется в».

Класс «Визуализация» описывает варианты представления процесса моделирования задачи - данный класс связан с классом «Постпроцессоры». Основываясь на схеме вычислительного эксперимента, в классе «Инструментальные средства» описаны программы-исполнители, разбитые на подклассы:

- препроцессоры – отвечают за расчетную сетку,
- решатели – реализуют основной вычислительный процесс, в частности – с использованием методов конечных элементов,
- постпроцессоры – отвечают за визуализацию процесса моделирования и конечных результатов,
- конвертеры данных – преобразовывают данные из одного формата в другой, если такая необходимость возникает при передаче данных между различными исполнителями.

Постановка задачи вывода на онтологии заключается в следующем: на основе сформированного исследователем в интерактивном режиме описания задачи строится запрос в виде совокупности формальных характеристик и ограничений на возможные значения атрибутов искомым объектам (рисунок 2).

Заданные в онтологии отношения между объектами классов позволяют исследователю задавать в явном виде неполную спецификацию решаемой задачи, а также избежать некоторых концептуальных ошибок при постановке задачи. Например, при указании конкретного сепарационного аппарата для моделирования, в результате вывода на онтологии может быть получено множество экземпляров руд, «подходящих» к данному аппарату. Выбор руды для данного варианта вычислительного эксперимента может осуществляться только из этого множества.

Далее, по заданной спецификации идентифицируются объекты различных классов в онтологии путем сопоставления с существующими объектами. Под идентификацией объектов будем понимать совокупность действий направленных на нахождение экземпляров интересующих объектов. В случае если идентификация объекта-исполнителя успешна, т.е. найден объект класса «Инструментальные средства» онтологии соответствующий искомому, проводится вывод экземпляров идентифицированного объекта с учетом заданных пользовательских ограничений на значения свойств. Эти экземпляры затем добавляются в класс «Исполнители».

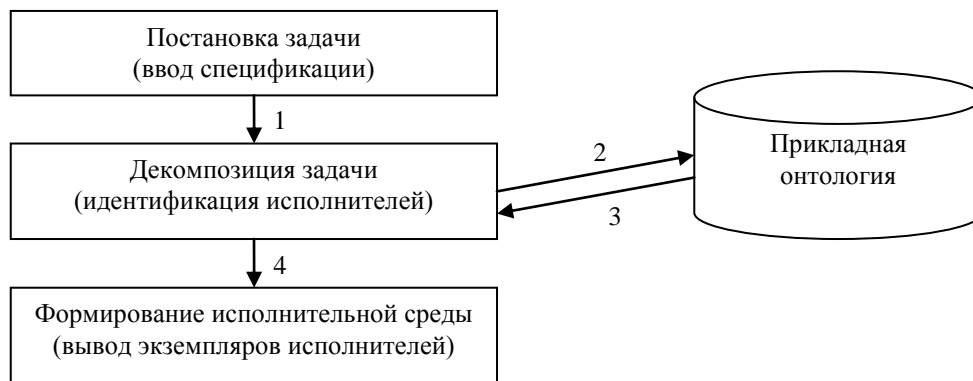


Рис. 2. Схема поиска исполнителей

Последовательность действий направленных на поиск множества исполнителей представлена ниже:

- 1) Задание спецификации исходной задачи вычислительного эксперимента:
  - маркировка экземпляров объектов онтологии (указание в спецификации определенных (конкретных) составляющих элементов задачи),
  - идентификация в онтологии объектов, соответствующих заданной спецификации,
- 2) вывод экземпляров объектов-исполнителей с учетом ограничений на значения свойств.

Для организации идентификации объектов в онтологии введем понятие запроса как сформированную исследователем структурно-логическую схему задачи с установленными ограничениями на значения свойств экземпляров объекта.

*Определение 5.* Пусть  $p_q$  – символьное имя свойства,  $r_q$  – ограничение на значение:

$$(5) \quad r = \langle c, v \rangle$$

где

$c$  – условие вывода ( $>$ ,  $<$ ,  $=$ ),

$v$  – значение свойства.

Тогда запрос  $Q$  к онтологии на идентификацию объектов класса  $N$  с заданными ограничениями можно задать формулой:

$$(6) \quad Q = \langle N, \langle p_0, \langle c_0, v_0 \rangle \rangle, \dots, \langle p_n, \langle c_n, v_n \rangle \rangle \rangle$$

После успешного нахождения всех требуемых исполнителей из них формируется спецификация исполнительской среды в простом случае в виде цепочки. В случае наличия альтернативных вариантов спецификации исполнительской среды возникает вопрос выбора той или иной альтернативы. Критерии выбора могут быть обусловлены различными факторами: ограничения на время решения задачи; условия доступа к инструментальным средствам; точность получения результата и др.

### 3. Построение графа исполнителей

Для решения задачи выбора альтернативы рассмотрим граф, вершины которого соответствуют найденным в процессе вывода на онтологии исполнителям, а дуги – связям между ними. Связи интерпретируются как абстрактные каналы передачи данных, отображающие возможность взаимодействия инструментальных средств. Направление передачи данных между исполнителями отражается направлением дуги, соединяющей соответствующие вершины графа (рисунок 3).

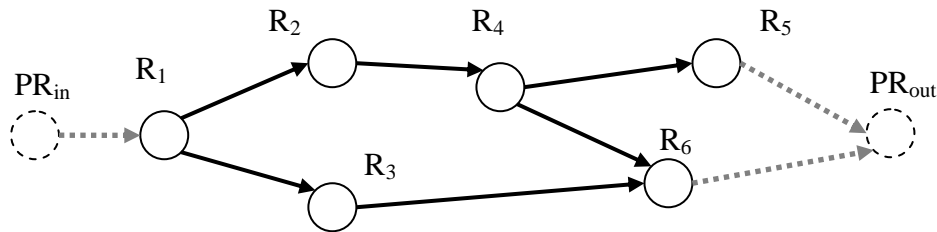


Рис. 3. Ориентированный граф исполнителей вычислительного эксперимента

Для оценки варианта решения, задаваемого некоторой цепочкой узлов, необходимо учитывать характеристики исполнителя, которые можно представить как «стоимость» его использования. Каждой дуге приписывается вес, зависящий от характеристик (сложность, точность) исполнителя, из которого эта дуга выходит, и от способа связи (наличие конвертера). Явно представлять конвертеры на графе нет необходимости. «Стоимостные» характеристики конвертера суммируются с характеристиками исполнителя, для которого данный конвертер требуется.

Интегральная стоимость представляет сумму всех числовых характеристик исполнителя с учетом ограничений спецификации задачи. Такие ограничения выражаются тем, что для каждой характеристики указывается ее значимость в виде баллов (например, от 1 до 5). Для оценок же используется уже нормализованная значимость:

$$(7) \quad w_i = \frac{S_i}{\sum_{j=1}^n S_j}$$

где:

$w_i$  – коэффициент значимости  $i$ -ой характеристики,

$S_i$  – заданная в спецификации значимость характеристики,

$n$  – число характеристик инструмента.

Тогда интегральную стоимость  $Value_{int}$  можно представить как сумму произведений значения характеристики  $p_i$  и коэффициента значимости этой характеристики  $w_i$ :

$$(8) \quad Value_{int} = \sum_{i=1}^n p_i \cdot w_i$$

Кроме того, спецификация задачи содержит описание исходных данных (входных данных) и требуемых результатов (выходных данных). Поэтому для построения цепочки решения в граф необходимо включить еще две вершины – два псевдоисполнителя.

Входной псевдоисполнитель  $PR_{in}$  содержит на выходе данные, идентичные исходным данным задачи. Выходной псевдоисполнитель  $PR_{out}$  должен получать на вход данные, идентичные требуемым результатам. Дуги, выходящие из входного псевдоисполнителя, имеют вес равный нулю, а дуги, входящие в выходной – веса в соответствии с характеристиками исполнителей, от которых они идут.

#### 4. Схема поиска альтернативных путей

В качестве основы алгоритма поиска альтернативных путей в графе исполнителей используется существующий алгоритм поиска кратчайшего пути на графах - алгоритмом Дейкстры [11] с учетом ограничений, описанных в спецификации исходной задачи. Такие ограничения могут касаться как характеристик инструментальных средств, связанных с ограничениями предметной области или ограничениями по точности, так и режима решения задачи: исследовательский (время не критично, но точность важна) или режим системы реального времени (время критично).

Поиск пути начинается в вершине, соответствующей входному псевдоисполнителю. Переходя по дугам, каждой вершине будем приписывать метки, представляющие пару:

$$(9) \quad M = \langle Class, Value \rangle$$

где

*Class* – класс исполнителя, от которого перешли по дуге;

*Value* – стоимость пути от исходной вершины до предыдущего исполнителя плюс вес дуги, по которой перешли.

Такой вид метки требуется для сохранения стоимости всех путей, так как исполнители из разных классов не являются альтернативными. При этом сохраняются сами пути с возможными параллельными участками. В итоге, необходимо пройти по всем дугам из начальной вершины и хотя бы раз посетить вершину, соответствующую выходному псевдоисполнителю.

Для примера рассмотрим случай (рисунок 4) со следующими обозначениями:

$R_i$  – исполнитель,

$C_j$  – класс исполнителя:

$$(10) \quad \begin{array}{ll} R_1 \in C_1 & R_2 \in C_2 \\ R_3, R_4 \in C_4 & R_5, R_6 \in C_3 \end{array}$$

Стоимость пути до исполнителя  $R_1$  равна 10. Выходные данные исполнителя  $R_1$  необходимо обработать двумя различными методами. Поэтому дальнейшее решение задачи может быть выполнено с помощью исполнителя  $R_2$  и одного из двух альтернативных исполнителей  $R_5$  и  $R_6$ . Последние принадлежат к одному подклассу  $C_3$  и поэтому, когда в ходе обхода графа попадаем в вершину  $R_7$ , у которой уже существует метка с подклассом, совпадающим с подклассом предыдущей вершины (например, перешли от  $R_6$ , а уже занесена метка от  $R_5 - \langle C_3, 16 \rangle$ ), выбирается наименьшая стоимость из двух. Пунктирной линией показано, что  $R_5$  хуже  $R_6$  по стоимости использования.



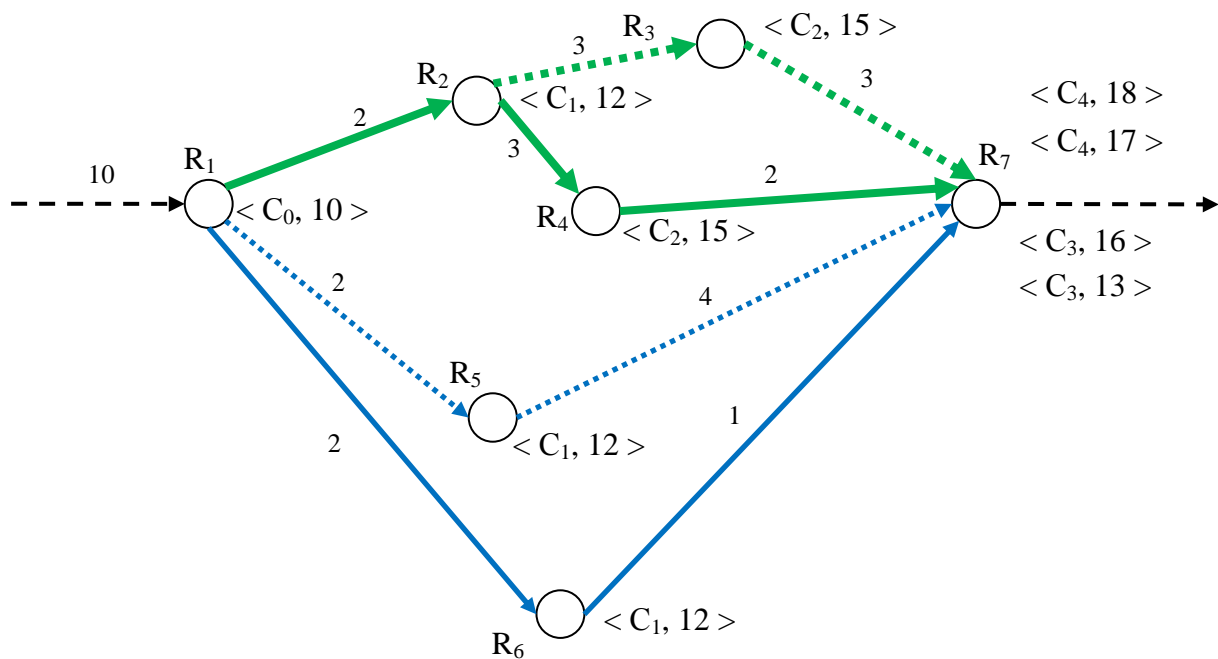


Рис. 4. Пример альтернативных путей в графе

За исполнителем  $R_2$  тоже появляется две альтернативы –  $R_3$  и  $R_4$ , но они уже из другого подкласса. При переходе в  $R_7$  необходимо рассматривать метки соответствующего класса и выбирать наименьшую стоимость -  $R_3$  хуже  $R_4$  (показано пунктиром).

В результате, стоимость использования каждого из построенных вариантов спецификации исполнительской среды вычисляется как сумма весов дуг, соединяющих исполнители, входящих в данную спецификацию. Такая стоимость будет учитывать данные из спецификации исходной задачи и из спецификаций инструментальных средств. Сравнение альтернатив в таком случае сводится к сравнению стоимостей их использования. Также на основании информации об используемых инструментальных средствах можно определить основные характеристики решения задачи – количество исполнителей, общее время выполнения, итоговая точность решения и т.п.

## 5. Заключение

Развитие программных и аппаратных средств компьютерного моделирования обеспечивает возможность их использования при решении обладающих высокой вычислительной сложностью задач совершенствования обогатительных технологий, процессов и аппаратов. При определенных условиях актуальной может оказаться проблема поиска наиболее эффективных вариантов реализации моделирования, и, как следствие, сравнительный анализ возможных альтернатив. Создание формального описания задач компьютерного моделирования обогатительных процессов и аппаратов в виде онтологии позволяет автоматизировать решение этой проблемы. Онтология обеспечит интеграцию и структурированное представление знаний различных исследователей в данной области, что в сочетании с механизмами вывода на онтологии существенно облегчает исследователю поиск возможных вариантов организации

моделирования и выбор варианта, наилучшим образом удовлетворяющего требованиям, предъявляемым в каждом конкретном случае.

#### СПИСОК ЛИТЕРАТУРЫ

1. *Олейник А.Г., Рыженко А.А. и др.* Информационно-аналитическое обеспечение комплексного управления горнопромышленными предприятиями // Информационные ресурсы России. 2005. № 4. С. 19-22.
2. *Емельянов С.В., Олейник А.Г. и др.* Информационные технологии регионального управления. М.: Едиториал УРСС, 2004.
3. *Бирюков В.В., Олейник А.Г.* Применение системы Femlab для моделирования гидродинамики течений в обогатительных аппаратах // Информационные ресурсы России. 2007. № 3 (97). С. 30-32.
4. *Неведров А.С., Олейник А.Г.* Об инструментальных средствах определения эффективных режимов обогащения минеральных руд // Информационные ресурсы России. 2011. №5 (123). С. 35-38.
5. *Неведров А.С., Олейник А.Г.* Формальное представление задачи интеграции инструментальных средств определения эффективных режимов обогащения минеральных руд // Труды Кольского научного центра РАН. Информационные технологии. Апатиты, Изд-во КНЦ РАН, 2011. Вып.2. № 4/2011(7). С. 196-202.
6. *Гладун А.Я., Рогушина Ю.В.* Онтологии в корпоративных системах. // Корпоративные системы. 2006. №1. С. 22-41.
7. *Смирнов С.В.* Онтологии в прикладных интеллектуальных системах: прагматический подход // Тр. 9-й национальной конференции по искусственному интеллекту с международным участием. 2004. Т. 3. С. 1059-1067.
8. *Жыжырий Е.А., Щербак С.С.* Математическое обеспечение систем поиска, основанных на онтологиях // Восточно-Европейский журнал передовых технологий. 2006. № 3. С. 12-17.
9. *Артемьева И.Л., Рештаненко Н.В.* Модульная модель онтологии органической химии // Интеллектуальные системы. 2004. № 2. С. 98-109.
10. *Кравец А.Г., Титова О.В.* Онтология физической задачи // Прикаспийский журнал: управление и высокие технологии. 2011. № 4(16). С. 12-16.
11. *Емеличев В.А., Мельников О.И. и др.* Лекции по теории графов. М.: Наука, 1990.