

УДК 004.75

А.В. ПРОХОРОВ, Е.Н. ВЛАДИМИРСКАЯ

*Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина***ЛОГИКО-КАТЕГОРНЫЙ ПОДХОД В ОНТОЛОГИЧЕСКОМ МОДЕЛИРОВАНИИ РАСПРЕДЕЛЕННЫХ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ**

Предложен подход автоматизированного преобразования моделей этапов разработки распределенной интеллектуальной системы с целью обеспечения адаптивности и высокого уровня модифицируемости. Описаны основные особенности объединения онтологических моделей с использованием теории категорий. Показаны преимущества использования категорного подхода для решения задач объединения онтологий. Описаны морфизмы, используемые при объединении онтологических моделей. Предлагается переход к многозначному исчислению присутствия для обработки неопределенных и противоречивых знаний при объединении онтологий.

Ключевые слова: мультиагентная система, интеллектуальный агент, онтология, OWL, теория категорий, морфизм.

Введение

В настоящее время одной из наиболее актуальных проблем является построение распределенных систем управления, обладающих необходимыми интегративными и интеллектуальными свойствами для принятия эффективных решений в условиях изменяющейся гетерогенной информационной среды. Решение этой проблемы на сегодняшний день заключается в эффективном совместном использовании дополняющих друг-друга технологий: мультиагентных систем (МАС)[1], сервис-ориентированной архитектуры и семантического веб. Важнейшим элементом таких распределенных интеллектуальных систем являются онтологии, которые позволяют соответствующим программным компонентам – интеллектуальным агентам – автоматически производить взаимодействие и обработку семантики информационных ресурсов и сервисов сети Интернет для эффективного решения различных прикладных задач.

Создание и внедрение распределенных интеллектуальных систем, и их широкое распространение выдвинуло проблему совершенствования методологии их создания на передний план. В условиях неполноты и несогласованности начальных требований к системе актуальными становятся задачи обеспечения гибкости, динамичности и легкого внесения изменений в разрабатываемую архитектуру, возможности изменять спецификации и пересоздавать реализацию системы, что влечет за собой необходимость автоматизации процессов создания и поддержки распределенных интеллектуальных систем. В работе предлагается подход к созданию МАС, особенностью которого является построение так называемой виртуальной онтологии, формируемой на основе несколь-

ких системных онтологических моделей. При этом чтобы иметь возможность производить операции и учитывать эволюционные изменения в онтологиях рассматривается математический аппарат теории категории [2], который выступает в роли универсального формализма онтологического описания и является независимым от выбора конкретного языка описания онтологий и специфики реализации. Действительно, в определении категории фиксируется некоторая совокупность объектов, или, по иной терминологии, доменов, которые связаны отношениями, или морфизмами. Отметим, что задачи взаимодействия двух онтологий в рамках такого представления ставятся и реализуются естественным образом через определение морфизмов из одной модели в другую.

Анализ последних исследований и публикаций

На данный момент разработано множество методологий [3, 4], которые объединяют процессы разработки МАС в единую цепочку. Общей идеей таких методологий является использование некоторой метамодели для создания МАС, включающей наиболее общие понятия и абстракции. К методологиям, основанным на объектно-ориентированных методах и технологиях с использованием соответствующих расширений, относится AUML [5]. Однако объектно-ориентированная модель не может охватить всю семантику, предоставляемую при использовании онтологий в МАС. Подход, использующий методологию MASE [6], в котором предлагается автоматизация преобразования моделей этапа анализа в модели этапа проектирования, также имеет ограничения, связанные с отсутствием интеграции с

онтологическими моделями. Методология Gaia [3], основанная на концепции роли за счет набора своих концептуальных моделей хорошо справляется с задачей агентно-ориентированного анализа предметной области, но обладает рядом существенных ограничений: интеллектуальная однородность агентов, статичность организационной структуры системы. В работах [7, 8] предлагаются основанные на OWL-онтологиях подходы проектирования и разработки МАС с элементами автоматизации некоторых этапов. Однако эти работы и разработанный инструментарий жестко связаны с областью медицинских проектов, что не позволяет их использовать как общую основу проектирования и разработки МАС.

Использование онтологического подхода при создании МАС в условиях обеспечения гибкости, динамичности и легкого внесения изменений в разрабатываемую архитектуру делает важным также вопрос, связанный с разработкой механизмов автоматизации объединения онтологических моделей, описывающих различные аспекты МАС. Перспективным представляется использование категорного подхода, обеспечивающего решение задач интеграции и отображения онтологий [10 – 13]. Достаточно детально категорный подход описывается в обзоре онтологических отображений [14]. Группировка онтологий также тесно связана с согласованием схем. В частности в [15] описывается алгебра согласования схем, а отображения онтологий и схем в общем описаны в [16, 17]. Необходимо отметить, что в настоящее время существуют готовые программные решения для некоторых операций объединения онтологических моделей. Одним из наиболее популярных является Jena [18], где поддерживаются базовые операции объединения онтологий. Отдельные возможности объединения онтологий также обеспечивает редактор онтологий Protégé и некоторые расширяющие его возможности модули [19].

Проведенный анализ, позволил выделить актуальное в настоящее время направление исследований, которое будет рассмотрено в данной работе, связанное с автоматизацией задач интеграции и отображения онтологических моделей при создании МАС, что обеспечивает, с одной стороны, автоматизированный синтез агентов системы, и с другой, дальнейшее эффективное использование онтологий агентами при взаимодействии и принятии решений.

Основные особенности разработки МАС и синтеза агентов на основе OWL-онтологий

Подход к автоматизации создания МАС изложен в [21] и включает в себя синтез агентов с добавлением им соответствующих механизмов взаимодействия, основанных на онтологиях. В данной ра-

боте будут рассмотрены особенности, связанные с первым шагом при реализации автоматизированного синтеза агентов системы – необходимых формализованных преобразований, что позволяет генерировать модели проектирования на основе аналитических моделей. Перед определением конкретных преобразований, будет описано, каким образом модели, полученные на этапе анализа, отображаются в модели этапа проектирования.

Онтология представляет собой таксономию понятий, расширенную некоторыми правилами-аксиомами. Эти правила специфицируются на некотором языке представления знаний, а агенты могут использовать эти знания для логического вывода. Для описания знаний выбран язык веб-онтологий OWL [22, 23]. Для автоматизации разработки МАС требуется объединение разработанных системных онтологий, которые являются повторно используемыми для различных предметных областей и проектов, с онтологическими моделями, специфичными для конкретной предметной области. Для удобства и обеспечения гибкости внесения изменений предлагается логическое разделение системных онтологий. Можно выделить следующие основные системные онтологии для построения МАС:

- модель знаний агента;
 - модель взаимодействия агентов в МАС и с внешними ресурсами;
 - онтология платформы JADE (эта агентная платформа была выбрана нами в силу своих преимуществ перед аналогичными), которая описывает элементы, специфичные для платформы построения агентов, а также элементы методологии построения МАС.
- На этапе анализа добавляются модели, описывающие основные аспекты предметной области и взаимодействия агентов, а именно:
- онтология вариантов использования, которая содержит информацию о пользователях системы, и предоставляемых им возможностям;
 - онтология предметной области (в работе это мультиагентная система управления контекстной рекламой в Интернет).

На этапе проектирования осуществляется автоматизированное установление отображений между элементами модели знаний и онтологии предметной области, а также между элементами модели взаимодействия и онтологии вариантов использования. Из модели вариантов использования извлекается информация о модели ролей и шаблонах поведения агентов.

В случае возникновения проблем или недостатка информации производится уточнение отображений и ролей (вручную). Основным результатом этапа проектирования является виртуальная онтология. Для построения виртуальной онтологии разра-

ботан также набор правил, описанных с использованием языка правил SWRL (Semantic Web Rule Language) [24]. SWRL позволяет использовать дизъюнкты Хорна (Horn-like rules) для явного указания способа вывода новых фактов из RDF-утверждений. Используя разработанные правила, виртуальная онтология пополняется новыми фактами.

Таким образом, в процессе преобразований формируется виртуальная онтология, которая связывает все другие онтологические модели с использованием механизмов отображения онтологий и представляет собой расширения модели знаний агентов элементами и связями других онтологий, а также служит основой для синтеза агентов. Основные связи между используемыми онтологическими моделями показаны на рис. 1.

Основные механизмы объединения онтологий и особенности построения виртуальной онтологии описаны в данной работе. Виртуальная онтология в дальнейшем служит основой для генерации кода агентов, а также для генерации и компоновки ресурсов, касающихся модели знаний. В результате генерируется совокупность агентов для MAC, которая служит основой для окончательной реализации системы.

При внесении изменений в модели предметной области и вариантов использования происходит изменение виртуальной онтологии и генерация новой версии системы. Автоматизация построения виртуальной онтологии обеспечивает более быстрый и легкий переход к расширению задач MAC и функциональных возможностей агентов.

Особенности построения виртуальной онтологии

Объединение моделей в виртуальную онтологию предполагает существования связей между элементами исходных онтологий. Самым простым типом связи является эквивалентность элементов. Объединение онтологий предполагает образование новой онтологии, которая идентифицирует эквивалентные элементы и при этом оставляет несвязанные элементы отдельными [2].

Для описания моделей используем следующие обозначения: agKM – онтологическая модель зна-

ний агента; agIM – онтологическая модель взаимодействия агентов; domCA – модель предметной области (управление контекстной рекламой – contextual advertising); usecaseCA – модель вариантов использования (для управления контекстной рекламой).

Подробную информацию о моделях можно найти в [21].

Определение 1. Класс **ОбК** множество объектов категории соответствует множеству понятий или классов в онтологии.

Отношение, которое описывает связи и представляет собой онтологическое отображение, для моделей agIM и agKM описывается следующим образом (рис. 2):

$$R_{virt1} \subseteq agKM \times agIM. \quad (1)$$

В простейшем случае представляет собой набор из пар эквивалентных элементов:

$$R_{virt1} = \{(Object, HostService), (Event, Error), (EventConflict)\}. \quad (2)$$

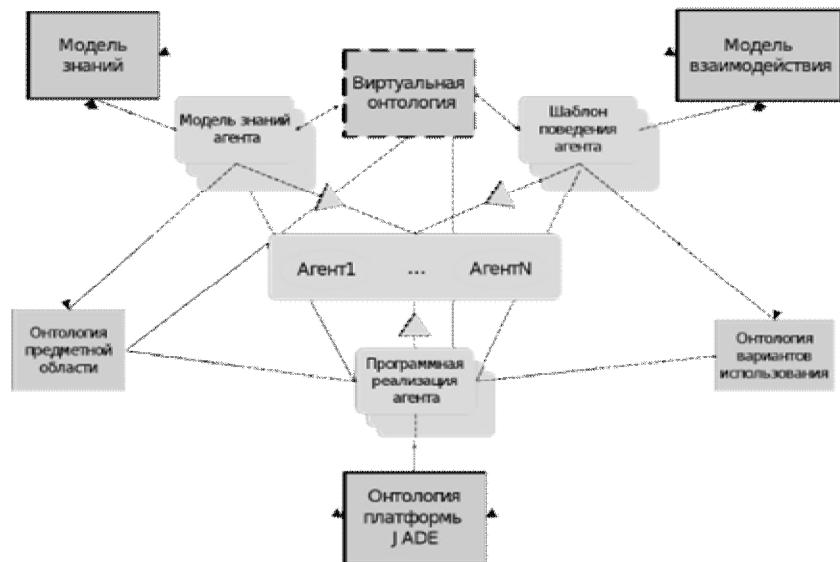


Рис. 1. Онтологические модели для синтеза агентов

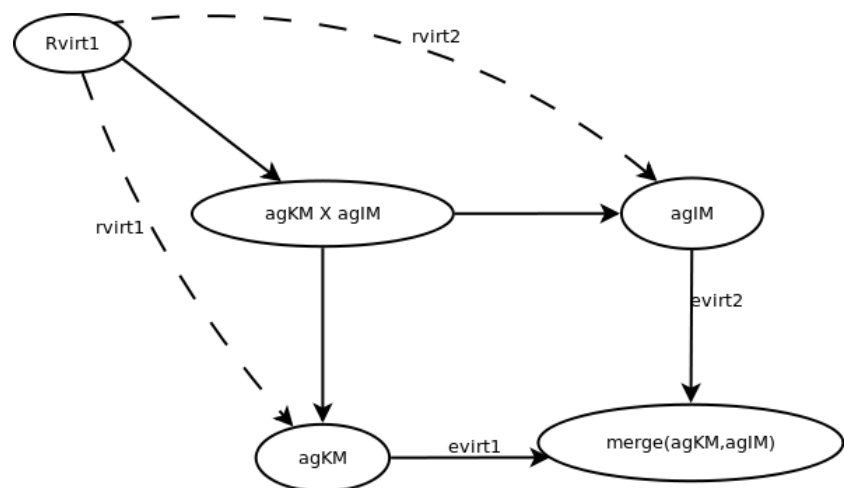


Рис. 2. Коммутативный квадрат для моделей agKM и agIM

Результат объединения agIM и agKM представляет собой кокартов квадрат [25] и удовлетворяет следующим условиям:

$$e_{virt1} \circ p_{virt1} = e_{virt2} \circ p_{virt2}; \quad (3)$$

$$\exists ! m : S \rightarrow T : f_1 = e_1 \circ m; \quad (4)$$

$$f_2 = e_2 \circ m.$$

Условие (4) верно для каждого объекта T и морфизмов $f_1 : P \rightarrow T$ и $f_2 : Q \rightarrow T$, где $f_1 \circ p_1 = f_2 \circ p_2$.

Модели rvirt1 и rvirt2 отображают каждую пару элементов R в первый и второй элемент пары соответственно. Аналогичным образом можно определить онтологические отображения для всех пар моделей:

$$R_{virt2} \subseteq agIM \times domCA;$$

$$R_{virt2} = \{(Object, AccessTool)\};$$

$$R_{virt3} \subseteq agKM \times domCA; \quad (5)$$

$$R_{virt3} = \{(Managed Resource, SiteObject); (HostName, SiteValuePartition)\}.$$

Для того чтобы объединить все полученные связи используются операции группировки и композиции группировок [2, 26]. На рис. 3 представлена композиция группировок.

$$\begin{aligned} R_{virt4} &\subseteq agKM \times domCA; \\ \langle R_{virt4}, rvirt41, rvirt42 \rangle; \\ rvirt41 &= rvirt11 \circ rvirt12; \\ rvirt42 &= rvirt21 \circ rvirt22. \end{aligned} \quad (6)$$

R_{virt4} определяет связи между онтологиями agKM и domCA, полученные в результате композиции связей между agKM и agIM, а также agIM и domCA.

Операция композиции [1,24] является предпочтительной, так как при композиции длинных цепочек связей через композицию пар смежных связей, порядок композиции не влияет на результат. Таким образом, получаем направленный граф онтологий и компонуемых связей между ними. С некоторыми

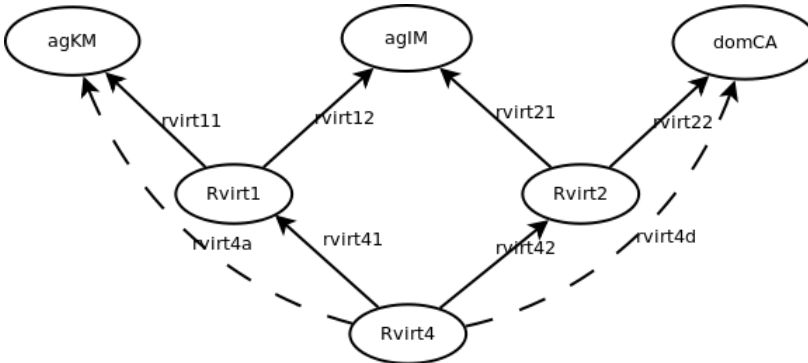


Рис. 3. Композиция группировок для моделей agKM и agIM, а также agIM и domCA

дополнительными посылками такой граф представляет собой категорию в математике, а связи называются морфизмами. Кроме того, между agKM и domCA существуют связи, не касающиеся agIM:

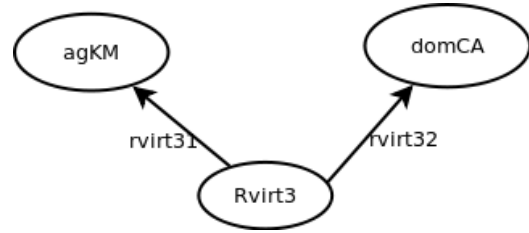


Рис. 4. Группировка моделей agKM и domCA

Чтобы учесть связи, определенные в и используем операцию объединения группировок. Объединение группировок - это кокартов квадрат пересечения группировок. Пересечение группировок определяется, как показано на рис. 5.

Пересечение предоставляет взаимные соответствия друг группировок. Объединение собирает все связи, определенные в двух группировках.

Далее подход расширяется и детализируется понятиями морфизма, композиции и W-группировок. Во многих случаях две онтологии, которые необходимо сгруппировать, содержат концепты, для которых нет эквивалентных в другой онтологии, но при этом они тесно связаны между собой.

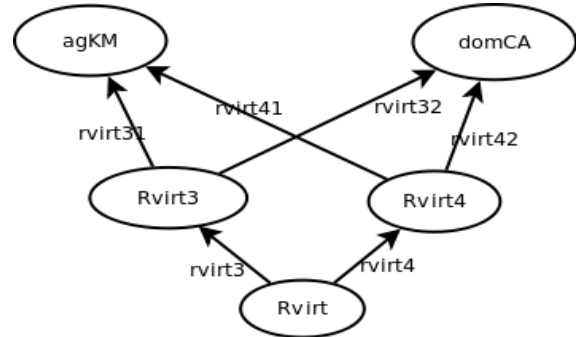


Рис. 5. Пересечение группировок для моделей agKM и domCA

Для определения связей, которые указывают не только на эквивалентность объектов (таких как owl:subClassOf, owl:disjointWith и др.) используется следующее определение.

Определение 2. Морфизм (отображение, стрелка) $f : A \rightarrow B$ в онтологии это тройка $\langle A, R, B \rangle$ такая, что:

- A и B это синтаксические сущности онтологии (в OWL это понятия или классы owl:Class, свойства rdf:Property, а также экземпляры из множества допус-

тимых значений определяемых доменом Dom)

– R определяет отношение, устанавливаемое между A и B.

В OWL в качестве таких отношений выступают свойства `rdf:Property`:

– свойства-объекты, как отношения между представителями двух классов

– свойства-значения, отношения между представителями классов и RDF-литералами или типами данных, определяемых XML Schema.

Для моделей `agKM` и `domCA` определяем следующие морфизмы (рис. 6):

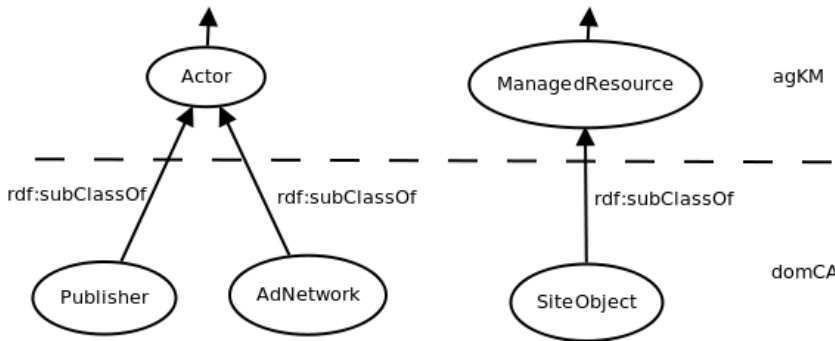


Рис. 6. Примеры морфизмов онтологии

```
<domCA:Publisher, rdfs:subClassOf,
agKM:Actor>,
<domCA:AdNetwork, rdfs:subClassOf,
agKM:Actor>,
<domCA:SiteObject, rdfs:subClassOf,
agKM:ManagedResource>
```

Таким образом, определяем категорию онтологий со связями `owl:subClassOf`, `equivalentClass`, `disjointClass`.

Определение 3. Ограничения отношений (`owl:Restriction`) в онтологии описываются в виде пары морфизмов $g:A \rightarrow D$, $h:D \rightarrow B$, где C это свойство некоторого класса B предполагая $f:C \rightarrow B$ (элемент `owl:onProperty`), A – некоторый класс накладывающий ограничения (например, класс возможных значений) на отношение или свойство C , D – ограничение имеющее отображение на класс, имеющий данное свойство C .

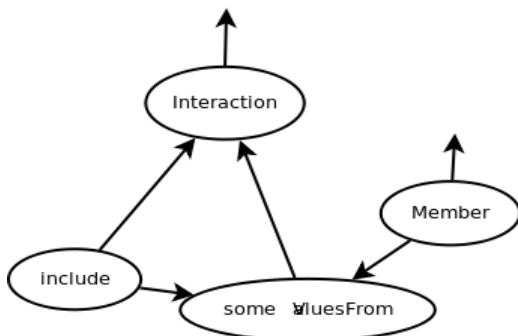


Рис. 7. Пример описания морфизмов онтологии

Операция композиции морфизмов определяется следующим образом (рис. 8):

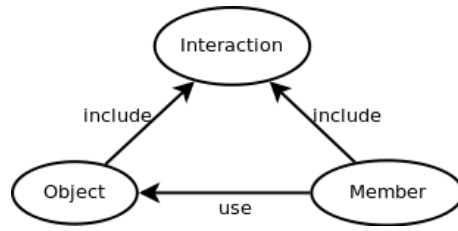


Рис. 8. Пример описания композиции морфизмов

Пусть $f:A \rightarrow B$ и $g:B \rightarrow C$ два морфизма в онтологии. Композицией морфизмов f и g в онтологии, обозначаемой $f \circ g$, является тройка $\langle A, R, C \rangle$ такая, что существуют $B, R1, R2$ такие, что имеются $\langle A, R1, B \rangle$, $\langle B, R2, C \rangle$ и $R = \varphi(R1, R2)$, где $\varphi: R \times R \rightarrow R$ и определяется по специальной таблице композиции.

Таким образом, морфизмы вместе с классами онтологии в качестве объектов образуют категорию.

При этом связи не ограничены конкретным языком онтологии [26].

Некоторые связи между онтологиями в предлагаемом подходе определены с использованием SWRL-правил [24]. Для объединения этой информации в виртуальную онтологию используются W-группировки (рис. 9).

Модель `swrlAg` – OWL/SWRL онтология, описывающая связи между `agKM` и `agIM`. SWRL-правила непосредственно не содержатся в указанных моделях, а только объединяют их. Результат объединения онтологий определяется, как показано на рис.10.

Промежуточные модели `agKM+` и `agIM+` представляют собой исходные онтологии `agKM` и `agIM`, расширенные элементами `swrlAg`, что позволяет описывать их как простую группировку V типа. Композиция группировок W типа является довольно простой в описании в отличие от полной алгебры группировок W типа, которая является довольно сложной) (рис. 11).

Онтология `rulesSWRL3` получается в результате объединения онтологий `swrlAg` и `swrlDom` в соответствии с W -группировкой $\langle agIM, R2, R3 \rangle$.

V -группировка `agKM` и `rulesSWRL`:

$$\langle R1, f1, r1 \circ g1 \rangle \tag{7}$$

V -группировка `agIM` и `rulesSWRL`:

$$\langle R4, g4, r3 \circ f4 \rangle \tag{8}$$

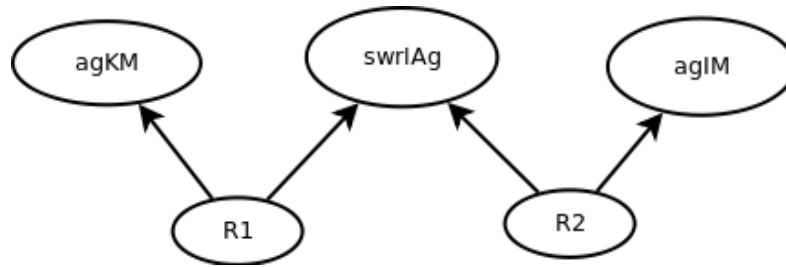


Рис. 9. W-групування для моделей agKM і domCA

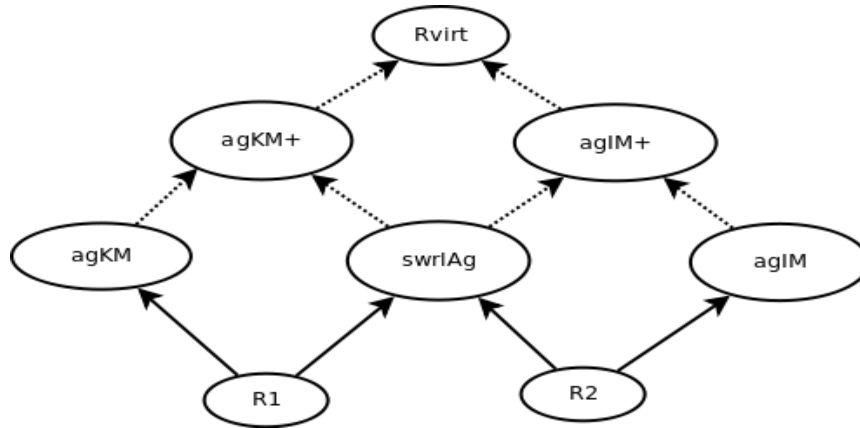


Рис. 10. Об'єднання з допомогою W-групувань

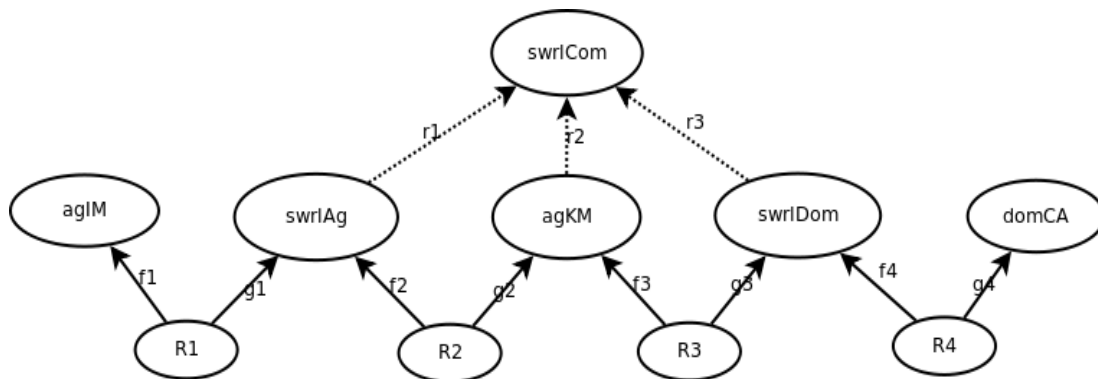


Рис. 11. Композиція W-групувань

Определение формализует связи между agIM и domCA через опосредованную онтологию agKM. Чтобы описать это одной промежуточной онтологией производится интеграция промежуточных моделей с agKM. Существует значительный недостаток: наследуя промежуточные аксиомы моделей swrlAg и agKM, в новую промежуточную онтологию смешивается вся вложенная информация. Но этот набор промежуточных аксиом может быть избыточным для достижения цели: он может включать аксиомы из модели agKM, которые не относятся ни к agIM, ни к domCA. Решение данной проблемы на определенном уровне заключается в решении задачи нахождения минимального избыточного набора аксиом, который приводит к набору релевантных результатов.

Общая схема преобразований при формировании виртуальной онтологии

Различные типы группировок имеют свои преимущества и недостатки. В то же время описанные подходы являются взаимодополняющими, и используя их комбинацию можно добиться наилучшего результата. Выделим основные моменты предлагаемого подхода комбинации группировок. Модель знаний тесно связана с онтологией предметной области. В этом случае мы используем V-группировку и детализируем категорию. Связь между agIM и agKM описываем с помощью промежуточной модели swrlAg и используем W-группировку. Таким образом, в swrlAg помещена информация промежуточного типа, которая не касается непосредственно группи-

руемых онтологий и описывается в основном с помощью правил SWRL. Аналогичным образом для моделей domCA и usecaseCA также используем

промежуточную модель swrlDom и W-группировку. Для формирования виртуальной онтологии используем операцию композиции группировок (рис. 12).

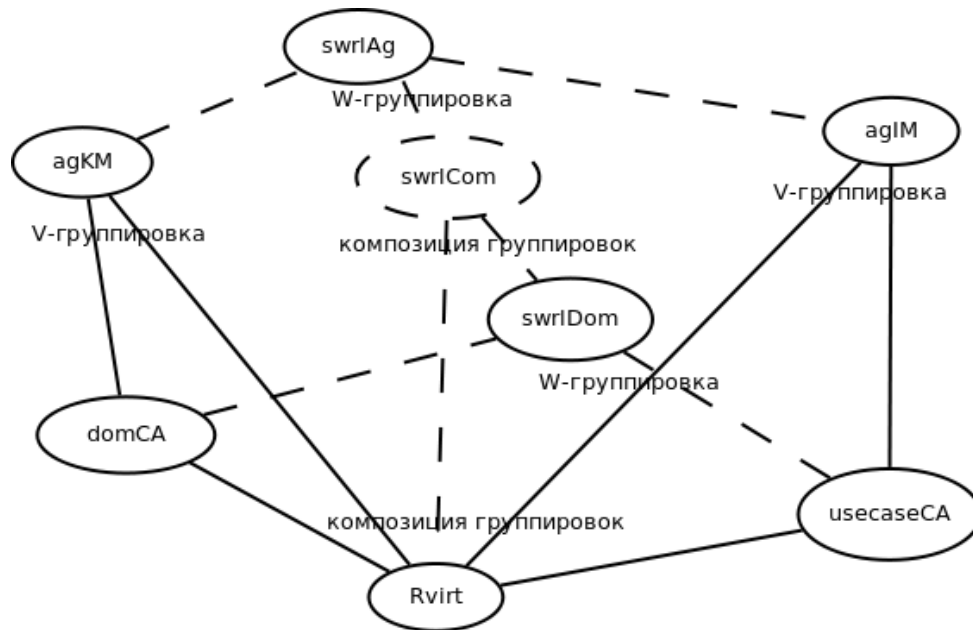


Рис. 12. Общая схема преобразований при формировании виртуальной онтологии

В результате применения всех преобразований получаем виртуальную онтологию **Rvirt**, которая является основой для генерации кода агентов мультиагентной системы.

Разработанные программные средства позволяют уточнять и корректировать процесс формирования виртуальной онтологии, добавляя возможность визуализации шагов преобразования в виде обновленного OWL-графа, с возможностью внесения изменений, кроме того, обеспечивается графическое представление категорий.

Важнейшей задачей при построении глобальной системы формализованных знаний является правильный выбор логической схемы. Разнообразные диалекты логических языков и соответствующие логические исчисления могут служить гибким инструментом для описания поведения объектов в рамках предметных областей. Для онтологий наибольшее распространение получила дескриптивная логика, которая является подмножеством логики предикатов первого порядка, ограниченным таким образом, чтобы гарантированно выполнялось любая процедура вывода (вычислительная разрешимость). Но с учетом рассмотренных особенностей работы с несколькими онтологиями важной задачей является также обеспечение возможности представлять в онтологиях неопределенности и работать с противоречивыми знаниями. Логические исчисления оказываются неприменимы в силу их неконструктивности

и экстенциональной теоретико-множественной основы. Альтернативой теории множеств как раз является логико-категорный подход, позволяющий рассматривать предметную область как сложную динамическую систему, состоящую из объектов различного характера, интерпретируемых на множестве их свойств. Кроме того, булевы топосы категории, как известно, могут использоваться для семантических конструкций логических теорий и именно в топосах получается очень естественно выразить многозначные логики. Поэтому в качестве основы логического исчисления предлагается использовать интенциональное многозначное исчисление присутствия [27]. Исчисление присутствия удовлетворяет требованиям, предъявляемым к логике процессов принятия решений и управления в распределенных интеллектуальных системах, предоставляет широкие возможности для описания связей по присутствию между объектами, их свойствами и атрибутами в предметной области, а также позволяет учитывать неопределенность и противоречивость знаний.

Заключение

Разработан метод синтеза агентов распределенной интеллектуальной системы на основе использования онтологических моделей знаний, позволяющий автоматизировано создавать новых агентов и менять поведение существующих для

обеспечения функционирования системы в условиях изменяющейся среды. В работе показаны преимущества использования категорного подхода для решения задач объединения онтологий. Разработана общая схема автоматизированной интеграции и объединения онтологических моделей в виртуальную онтологию. Описаны морфизмы, используемые при объединении онтологических моделей. Достаточный уровень поддержки автоматизации при преобразованиях позволяет проектировщику системы принимать решения только верхнего уровня, в то время как низкоуровневые детали преобразований выполняются автоматически. Система преобразования также обеспечивает возможность оперативного контроля и прозрачность процесса разработки системы, начиная с разработки требований и до генерации исполняемого кода. Разработанное программное обеспечение позволяет уточнять и корректировать процесс формирования виртуальной онтологии, а также предоставляет возможность отображения шагов преобразования в виде обновленного OWL-графа и графического представления категорий. Положения, изложенные в данной работе, легли в основу разработки метода представления знаний на основе OWL-онтологий с аксиоматическими теориями, использующими многозначное исчисление присутствия, что позволяет строить иерархические многоуровневые формализованные аксиоматические теории MAC, учитывать неопределенность и противоречивость знаний.

Литература

1. Гаврилова Т. А. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский. – СПб.: Пупер, 2003. – 215 с.
2. Clint Sparkman S. *Automated Derivation of Complex Agent Architectures from Analysis Specifications*, [Электронный ресурс] / Clint Sparkman Scott, Scott A DeLoach, and Athie L Self, *proceedings of the second international workshop on agent-oriented software engineering*, 2001. – Режим доступа: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.29.750>.
3. Wooldridge M. *The Gaia Methodology for Agent-Oriented Analysis and Design* [Электронный ресурс] / Michael Wooldridge, Nicholas R. Jennings, David Kinny // *Journal of Autonomous Agents and Multi-Agent Systems*, 2000. – Режим доступа: <http://www.ecs.soton.ac.uk/~nrj/download-files/jaamas2000.pdf>.
4. Scott A. DeLoach *The MaSE Methodology*. In *Methodologies and Software Engineering for Agent Systems. The Agent-Oriented Software Engineering Handbook Series : Multiagent Systems, Artificial Societies, and Simulated Organizations* / Bergenti, Federico; Gleizes, Marie-Pierre; Zambonelli, Franco (Eds.), Kluwer Academic Publishing. – 2004. – Vol. 11.
5. Odell H. Van Dyke Parunak. *Representing Agent Interaction Protocols in UML* / Odell, H. Van Dyke Parunak and B. Bauer // *Proceedings of Agents 2000*, 2000. – P. 126-138.
7. Hajnal A. *Ontology-Driven Agent Code Generation for Home Care in Protégé* / A. Hajnal, G. Pedone, LZ. Varga // *10th International Protege Conference*, 2007. – P. 91-93.
8. *An Ontology-Driven Framework for Deploying JADE Agent Systems* / CI. Nyulas, MJ O'Connor, Tu SW, DL Buckeridge, A Okhmatovskaia, MA. Musen // *IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'08)*, 2008.
10. Verheijen Danladi *Encapsulation and composition of ontologies* / Danladi Verheijen, Gio Wiederhold, Jan Jannink, Srinivasan Picha // *In Proc. 15th National Conference on Artificial Intelligence (AAAI'98)*, July 1998. – P. 45-48.
11. Kent Robert E. *The information flow foundation for conceptual knowledge organization.* / Robert E. Kent // *In Proc. 6th International Conference of the International Society for Knowledge Organization (ISKO)*, 2000. – P. 142-145.
12. Barwise John *Information flow: the logic of distributed systems* / John Barwisem, Jerry Seligman // *Cambridge University Press, Cambridge, United Kingdom*. – 1997. – Number 44. – P. 40-43.
13. *What is ontology merging? - a categorytheoretic perspective using pushouts* / Pascal Hitzler, Markus Krotzsch, Marc Ehrig, York Sure // *In Proc. First International Workshop on Contexts and Ontologies: Theory, Practice and Applications (C&O)*, AAAI Press, July 2005 - P. 104-107.
14. Kalfoglo Yannis *Ontology mapping: The state of the art* / Yannis Kalfoglou, Marco Schorlemmer // *In Semantic Interoperability and Integration*, number 04391 in *Dagstuhl Seminar Proceedings*, Dagstuhl, Germany, 2005. – P. 63-68.
15. Bernstein Philip A. *Applying model management to classical meta data problems* / Philip A. Bernstein // *In Proc. First Biennial Conference on Innovative Data Systems Research (CIDR'03)*, January 2003. – P. 127-144.
16. Rah Erhard *A survey of approaches for automatic schema matching* / Erhard Rahm, Philip A. Bernstein // *The Very Large Data Base Journal*. – December 2001. – 10(4). – P.334-350.
17. Shvaik Pavel *A survey of schema-based matching approaches* / Pavel Shvaiko and Jerome Euzenat // *Journal on Data Semantics*. – 2005. – 4. – P. 146-171.
18. *Операционные модели Jena* [Электронный ресурс]. – Режим доступа: http://jena.sourceforge.net/tutorial/RDF_API/index.html#ch-Operations%20on%20Models – Название с титул. экрана.
19. *Плагины Protege* [Электронный ресурс]. – Режим доступа: <http://protege.cim3.net/cgi-bin/wiki.pl?ProtegePluginsLibraryByType> – Назв. с титул. экрана.
20. *Jade programmer's guide* / Fabio Belfemine, Giovanni Caire, Tiziana Trucco, Giovanni Rimassa. –

TILAB, formerly CSELT, University of Parma – 2006.

21. Прохоров А.В. Автоматизированный синтез агентов при создании мультиагентных систем / А.В. Прохоров, Е.Н. Владимирская // Восточно-Европейский журнал передовых технологий. – 2009. – № 39(3/4). – С. 25-31.

22. Смит Майкл К. OWL, язык веб-онтологий. Руководство [Электронный ресурс] / Майкл К. Смит, Крис Велти, Дебора Л. МакГиннес. – Режим доступа: <http://www3.org/TR/2004/REC-owl-guide-20040210/>.

23. Laclavik Michal AgentOWL: Semantic Knowledge Model and Agent Architecture / Michal Laclavik, Marian Babik, Zoltan Balogh, Ladislav Hluchy // Computing and Informatics. – 2006. – Vol. 25, no. 5. – P. 419-437. – ISSN 1335-9150, – Chapters 1, 4, 5.

24. SWRL: A Semantic Web Rule Language Combining OWL and RuleML [Электронный ресурс] W3C

Member Submission 21 May 2004. – Режим доступа: <http://www.w3.org/Submission/SWRL/>. – Название с титул. экрана.

25. Голдблатт Р. Топосы. Категорный анализ логики / Р. Голдблатт. – М.: Мир, 1983. – 487 с.

26. Zimmermann. Formalizing Ontology Alignment and its Operations with Category Theory [Электронный ресурс] / Zimmermann // Proceeding of the 2006 conference on Formal Ontology in Information Systems: Proceedings of the Fourth International Conference 2006. – P. 277-288. – Режим доступа: <http://portal.acm.org/citation.cfm?id=1566111>.

27. Прохоров А.В. Методы и модели логистического анализа процессов принятия решений и управления автоматизированными многоуровневыми распределенными системами: дисс. канд. техн. наук: 05.13.06 / Прохоров Александр Валерьевич. – Харьков, 2003. – 309 с.

Поступила в редакцию 2.12.2010

Рецензент: д-р техн. наук, проф., зав. каф. информатики А.Ю. Соколов, Национальный аэрокосмический университет им. Н.Е. Жуковского «Харьковский авиационный институт», Харьков.

ЛОГИКО-КАТЕГОРИЙ ПІДХІД В ОНТОЛОГІЧНОМУ МОДЕЛЮВАННІ РОЗПОДІЛЕНИХ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ

О.В. Прохоров, К.М. Владимирська

Запропоновано підхід автоматизованого перетворення моделей етапів розробки розподіленої інтелектуальної системи з метою забезпечення адаптивності та високого рівня модифікуємості. Зображені основні особливості об'єднання онтологічних моделей з використанням теорії категорій. Наведені переваги використання категорного підходу для вирішення проблем об'єднання онтологічних моделей. Описані морфізми, які використовуються при об'єднанні онтологічних моделей. Запропоновано перехід до багатозначного числення присутності з метою обробки невизначених то суперечливих знань.

Ключові слова: мультиагентна система, інтелектуальний агент, база знань, онтологія, OWL, теорія категорій, морфізм.

CATEGORY APPROACH TO THE ONTOLOGICAL MODELING IN DISTRIBUTED INTELLIGENT SYSTEMS

A.V. Prohorov, C.N. Vladimirska

The approach to automatic conversion of the distributed intelligent system development stages models for the higher level adjustability and modifiability is introduced. The main features of the ontological models integration using category theory are described. The advantages of using category theory for the ontologies merging problem solving are showed. Morphisms used to merge ontological models are described. A conversion to the multiple-valued presense calculation for the uncertain and contradiction knowledge processing is discussed.

Keywords: multi-agent system, intelligent agent, knowledge base, Semantic Web, ontology, OWL, category theory, morphism.

Прохоров Александр Валерьевич – канд. техн. наук, доцент, доцент каф. информационных управляющих систем, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков, Украина.

Владимирская Екатерина Николаевна – аспирант каф. информационных управляющих систем, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков, Украина, e-mail: catherine.vladimirska@gmail.com.