

Д.Н. БИРЮКОВ, А.Г. ЛОМАКО
**ДЕНОТАЦИОННАЯ СЕМАНТИКА КОНТЕКСТОВ ЗНАНИЙ
ПРИ ОНТОЛОГИЧЕСКОМ МОДЕЛИРОВАНИИ ПРЕДМЕТНЫХ
ОБЛАСТЕЙ КОНФЛИКТА**

Бирюков Д.Н., Ломако А.Г. **Денотационная семантика контекстов знаний при онтологическом моделировании предметных областей конфликта.**

Аннотация. Предлагается подход к онтологическому описанию произвольной предметной области, основанный на применении концептов трёх типов: “Объекты”, “Свойства” и “Действия”. При этом различные аспекты представлений, используемые для описания знаний, предлагается частично упорядочивать свойством аппроксимации в функциональные слои, сегменты и области. Это должно позволить моделировать семантические особенности контекстно-зависимых знаний предметных областей, учитывать их изменения и уточнения при порождении решений. Это открывает возможности прогнозирования намерений и предотвращения реализации киберугроз критической информационной инфраструктуре.

Ключевые слова: знания, контекст, онтология, аппроксимация, память, денотационная семантика, конфликт, киберсистема.

Biryukov D.N., Lomako A.G. **Denotational Semantics of Knowledge Contexts in Ontological Modeling of Subject Domains of the Conflict.**

Abstract. An approach to the ontological description of any subject domain based on application of concepts of three types is offered: "Objects", "Properties" and "Actions". Thus, it is proposed to partially arrange various aspects of representations used for the description of knowledge by approximation property in functional layers, segments and areas. This makes possible to model semantic features of context-dependent knowledge of subject domains, to consider their changes and specifications at generation of decisions. Moreover, this opens the possibilities of predicting intentions and preventing realization of cyber threats to critical information infrastructure.

Keywords: knowledge, context, ontology, approximation, memory, denotational semantics, conflict, cybersystem.

1. Введение. Информационные и коммуникационные технологии используются для усиления человеческого интеллекта, дополняя его творческие способности в работе с информацией. Исследователи предсказывают, что по мере развития данной области будет происходить формирование «внешней коры» мозга («экзокортекса»), то есть, системы программ, дополняющих и расширяющих мыслительные процессы человека [1].

Очевидно, что чем более интеллектуальными будут искомые программы, тем большее количество задач в поиске решений сможет переложить на них человек, особенно в области информационной безопасности при защите критической информационной инфраструктуры от атакующих воздействий. Также можно предположить, что интеллектуальность проектируемых и разрабатываемых программ суще-

ственно зависит от их возможности осуществлять смысловую, связанную с контекстом, обработку знаний и данных. Приоритетной является разработка систем, способных осуществлять семантическую обработку поступающих и накопленных данных и знаний и синтезировать в начале возможные, а затем всё более точные и точные, контекстно-зависимые от условий задач, решения. Искомая система должна быть способной синтезировать сценарии упреждающего поведения в конфликте, что должно способствовать прогнозированию намерений и предотвращению кибервоздействий на КИИ.

2. Подход к моделированию контекстно-зависимых онтологий. Учитывая особенности организации данных в семантической памяти человека [2-5], целесообразно представлять знания о предметной области (ПрО) в виде иерархии структурированных объектов, связанных между собой отношениями. На этой идее базируются такие модели представлений, как фреймы, семантические сети, UML и т.д. К сожалению, все эти языки хоть и являются достаточно удобным средством для представления знаний о ПрО конфликта, но лишены возможностей отражения семантики, а выражаемая в них информация предназначена скорее для человеческого, а не машинного восприятия. Также, большое развитие получила логика предикатов (FOL), в которой присутствуют механизмы семантической обработки, но нет удобных средств представления знаний.

В 1985 году Р.Брахман предложил объединить семантические сети и FOL. То, что получилось, было названо терминологической логикой, а далее, в процессе развития, превратилось в семейство дескрипционных логик [6], которые нашли своё практическое применение при описании онтологий в развивающемся в настоящий момент семантическом Web-е.

Онтология – формальная спецификация концептуализации, которая имеет место в некотором контексте предметной области (Gruber, 1993). Онтология описывает основные концепции (положения) предметной области и определяет отношения между ними.

Концептуализация представляет собой описание понятий, а также всю информацию, имеющую отношение к понятиям (свойства, отношения, ограничения, аксиомы, утверждения), необходимую для описания и решения задач в избранной предметной области. Таким образом, онтология состоит из примитивов представления знаний о предметной области (определений основных понятий, таких как имена индивидуумов, классы, функции и другие сущности), а также различного рода семантических связей, поддерживаемых между ними.

Кроме того, под онтологией можно понимать формальное описание результатов концептуального моделирования предметной области, представленное в форме, удобной для восприятия человеком и при реализации в компьютерной системе [7].

Для описания онтологий используются соответствующие языки, например – OWL (Web Ontology Language). OWL обеспечивает три различных по выразительности диалекта (см. рисунок 1) [8]. Применение OWL может позволить распределённое хранение данных в структурированном виде.

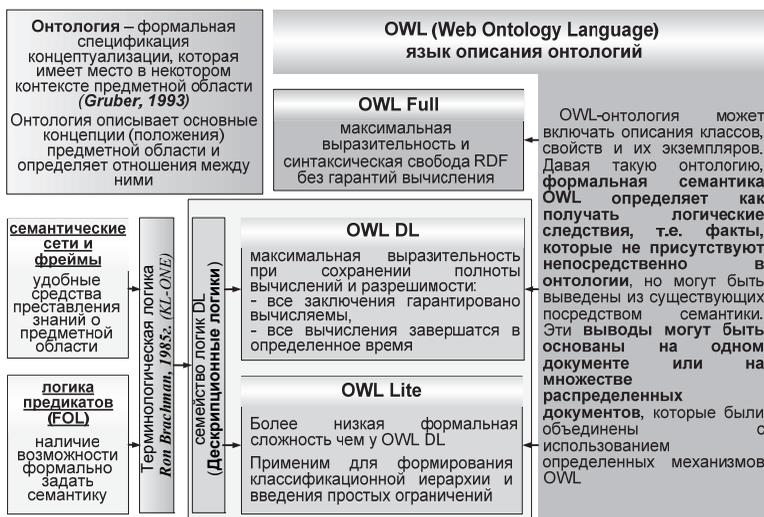


Рис. 1. Схема взаимосвязи семантических сетей, логики предикатов и OWL

На начальном этапе, в ходе построения онтологии, необходимо задать перечень Концептов (унарных предикатов) и Ролей (бинарных предикатов), использование которых позволило бы в дальнейшем описать Про конфликта с учётом возможных контекстов. При задании конкретных ролей, используемых при онтологическом моделировании, описание каждой роли может быть уточнено путём задания определённых характеристик. Основные характеристики (свойства) ролей: симметричность, функциональность, рефлексивность, транзитивность, антисимметричность, обратная функциональность, иррефлексивность. В зависимости от того, какие характеристики введены (используются), может применяться та или иная (с позиций выбранной логики) машина логического вывода.

Следует отдельно отметить, что сам онтологический подход к описанию произвольных ПрО не вносит определённых ограничений на порядок выбора и использования тех или иных типов Концептов и Ролей (кроме требований относительно правильного использования ролей с приведёнными выше характеристиками). Данный факт на практике приводит к тому, что инженеры по знаниям для описания разных ПрО используют большое количество различных Ролей, несущих различную семантику, что в лучшем случае не способствует, а чаще и вообще делает невозможным объединение различных онтологий (онтологий, описывающих различные ПрО и созданные различными специалистами). В тех же случаях, в которых такое объединение возможно, зачастую следует изменять машину логического вывода, используемую для порождения новых знаний над новой (объединённой) онтологией. Наличие же возможности объединения онтологий, описывающих различные ПрО конфликта, просто необходимо для того, чтобы проектируемая интеллектуальная система (ИС) имела потенциальную возможность к обучению, состоящему в переносе знаний о конфликтах и их разрешениях из одной ПрО в другую (например: из “биосферы” и “военного искусства” в “кибер-сферу”).

Для того, чтобы проектируемая система имела возможность синтезировать сценарии упреждающего поведения в конфликтах, она должна иметь возможность представлять и обрабатывать знания о причинах конфликтов, процессах их протекания и следствиях конфликтов, а для этого необходимо, чтобы она обладала возможностью представления знаний об объектах, их свойствах и процессах взаимодействия различных объектов и субъектов. При этом следует учесть, что сами рассматриваемые предметные области не должны вносить ограничений, приводящих к невозможности описания знаний из этих ПрО в единой онтологии. Так как в противном случае, это может привести к принципиальной невозможности обогащения ИС знаниями о поведении в конфликтах, имеющих место в других ПрО.

Достаточно важным вопросом при манипулировании знаниями, представленными посредством единой онтологии в Базе Знаний (БЗ) ИС, является вопрос, связанный с выделением из онтологии тех её фрагментов, знания из которых должны быть доступны системе в тот или иной момент времени при решении тех или иных задач, т.е. при различных контекстах – см. рисунок 2 (на рисунке каждый концепт представляет собой определённый тип действий (А); роль, отображённую пунктирной линией, следует интерпретировать как “является подклассом”, а роль, отображённую сплошной линией, - “следует за”). Если проводить аналогию с механизмами, протекающими в памяти человека, то данную задачу можно переформулировать как задачу управления Фокусом Внимания [9] и задачу помещения части данных в Эпизодический Буфер [10] для их дальнейшей обработки.

С учётом этого можно сформулировать ряд требований, которые должны быть учтены при создании системы метамоделирования, представления и манипулирования знаниями. Система метамоделирования знаний должна быть способной:

- представлять и обрабатывать знания из различных предметных областей об объектах, их свойствах и процессах (для возможности обогащения БЗ описаниями различных конфликтов, которые могут наблюдаться в различных ПрО);

- содержать ограниченное количество типов Концептов и Ролей, достаточное для описания произвольных ПрО (для задания единых правил построения онтологии, способствующих дальнейшему их объединению, а также применению единых логических правил для порождения новых знаний вне зависимости от специфики ПрО и рассматриваемых конфликтов);

- иметь разрешимые алгоритмы направленного поиска данных (для реализации возможности определения необходимого фрагмента онтологии исходя из задач и контекстов);

- осуществлять поиск подобия фрагментов онтологии по аналогии (для осуществления поиска по аналогии подобия возникающих задач и их решений в ходе конфликтов в разных ПрО);

- изменять доступность знаний для их дальнейшей обработки (для упорядочивания получаемых решений, определения наиболее ассоциируемых знаний, а также “забывания” ложных и мало используемых / мало подтверждённых знаний).

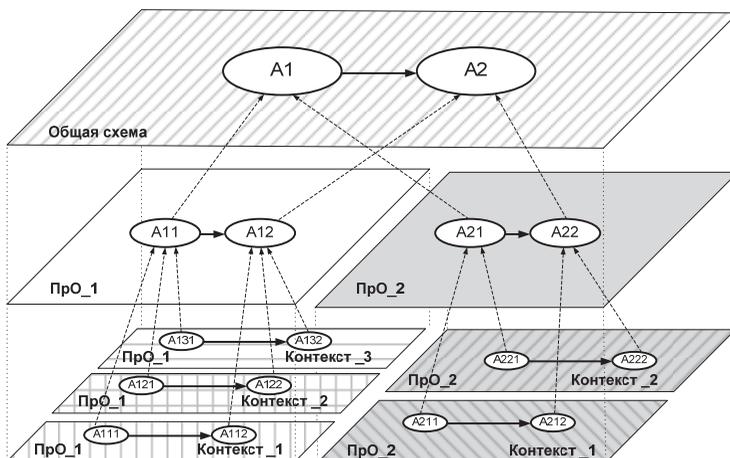


Рис. 2. Пример обобщенного представления контекстно-зависимых знаний предметных областей в единой онтологии

Искомые способности требуют формального определения денотационной семантики контекстов знаний при онтологическом моделировании предметных областей конфликта.

Вопрос, связанный с наделением языков программирования математической семантикой, был поднят ещё в 1971 году Д.Скоттом [11], однако и по сей день, предложенная им математическая теория не развита до уровня, позволяющего применить её для решения конкретных практических задач на должном уровне.

В качестве центрального элемента математической семантики, введённой Д. Скоттом, можно указать на абстрактные функции от элементов того типа данных, который связан с входными переменными, и принимающие значения элементов того типа данных, который связан с выходом. Очевидно, что при переходе к более “абстрактному” описанию решаемых задач на уровне функций и входных/выходных данных на уровне типов, невозможно полностью уйти от операциональных аспектов, поскольку в конечном итоге функции должны быть реализованы и выполняться на вычислительных средствах, способных манипулировать только конечными конфигурациями. В то же время, математическая семантика должна позволить осуществлять операциональное моделирование абстрактных сущностей.

Для того чтобы стало возможным манипулирование абстракциями, необходимо иметь соответствующий математически определенный язык. В работе [11] Д.Скотт предложил заслуживающий внимания взгляд на природу типов данных и функций (отображений) из одного типа данных в другой. Он также указал на то, что в математике недостаточно проработаны вопросы, связанные с описанием функций, определённых на всех допустимых функциях как аргументах и применимых даже к самим себе как к аргументу, и предложил математическую теорию функций, которая может быть использована как проект, дающий “корректный” подход к семантике.

Далее, в работе [12], Д.Скотт с помощью решёток в общих терминах описал абстрактную теорию конечной аппроксимации и бесконечных пределов. Эти понятия он применил для эффективного построения некоторого класса пространств (типов данных), в том числе и функциональных, что позволило использовать их как пространство математических значений в семантических интерпретациях языков программирования высокого уровня.

В статье [13] А.Шамир и У.Уэйдж, опираясь на формализмы, введённые Д.Скоттом, изложили новый подход к семантике типов данных, при котором типы сами включаются как элементы в область объектов. Этот подход позволяет типам иметь подтипы, позволяет рас-

смагивать подлинно полиморфные функции и дает точную семантику для рекурсивных определений типов (включая определения с параметрами). Кроме того, данный подход дает простые и прямые методы для доказательства типовых свойств рекурсивных определений.

Некоторые учёные [14-18] выступали за алгебраическое рассмотрение типов данных, при котором подчёркивался тот факт, что тип данных состоит не только из множества (иногда частично упорядоченного), но и из операций, удовлетворяющих некоторым равенствам.

А.Демерс, Дж.Донахью, Р.Гейтельбаум и Дж.Уильямс также указали [19] на то, что типы данных необходимо рассматривать неразрывно с операциями, реализуемыми на них: “Если два типа отличаются смыслом перечисленных операций, который задается определениями этих типов, то их следует считать различными”.

Таким образом, следует указать на наличие двух точек зрения на сами функции, а именно: экстенциональной и интенциональной точки зрения. При экстенциональном взгляде на функции они рассматриваются как абстрактные объекты, которые полностью определяются своими парами (аргумент, значение), т.е. своими графиками. Т.е. подразумевается выполнимость аксиомы экстенциональности, утверждающей, что если $f(x) = g(x)$ для всех x , то $f = g$.

Этой точки зрения противостоит интенциональная точка зрения, которая представляется очень существенной [20] в контексте вычислимости и программирования, согласно которой функция – это «операция», правило «преобразования» или «вычисления», какое-то определение такого правила.

Следует также отдельно остановиться на теории типов Ч.Хоара [21], в основе которой лежит понятие типа, отличительными особенностями которого являются следующие:

- 1) тип определяет класс значений, которые могут принимать переменная или выражение;
- 2) каждое значение принадлежит одному и только одному типу;
- 3) тип значения константы, переменной или выражения можно вывести либо из контекста, либо из вида самого операнда, не обращаясь к значениям, вычисляемым во время работы программы;
- 4) каждой операции соответствует некоторый фиксированный тип ее операндов и некоторый фиксированный тип результата;
- 5) для каждого типа свойства значений и элементарных операций над значениями задаются с помощью аксиом;
- 6) при работе с языком высокого уровня знание типа позволяет обнаруживать в программе бессмысленные конструкции и решать вопрос о методе представления данных и преобразования их в вычисли-

тельной машине;

7) интересующие нас типы - это типы, хорошо знакомые математикам: прямые произведения, размеченные объединения, множества, функции, последовательности и рекурсивные структуры.

Ряд положений, указанных Ч.Хоаром, спорны (например, положение 2, так как ряд авторов считают, что некоторые типы могут быть подмножествами других типов [22-24] или что объект может принадлежать конечному числу типов [25]. Этот вопрос обсуждается в работах [13, 26, 27] и является частью более общего вопроса о взаимоотношении между типами), однако положение, в котором указывается, что для каждого типа свойства значений и элементарных операций над значениями задаются с помощью аксиом – заслуживает отдельного внимания.

3. Предложения по представлению знаний для интеллектуальной системы. В работе [11] Д.Скотт пришёл к выводу о том, что пространство функций также должно рассматриваться как имеющее тип данных. Так как функция есть, вообще говоря, сама по себе бесконечный объект, то Д.Скотт предложил идею конечного приближения (аппроксимации). Он утверждал [28], что существует общая теория конечной аппроксимации и имеется много типов объектов, которые могут быть получены как пределы аппроксимации.

Математическая теория, предложенная Д.Скоттом, основана на идее, что *отношение аппроксимации* (\sqsubseteq) должно быть *частичным порядком*, и годится не любая структура с частичным порядком, а такая, в которой можно брать пределы.

Запись $x \sqsubseteq y$ означает, что x аппроксимирует y . Это качественное отношение, из которого должно следовать, что x совместимо с y . Можно утверждать, что x хуже, а y лучше (или наоборот), но нельзя сказать, как близко x находится по отношению к y .

В дальнейших рассуждениях принимается, что типы (по меньшей мере) структурированы отношениями \sqsubseteq . В соответствии с интуитивным пониманием следует считать, что отношение \sqsubseteq : рефлексивно ($\forall a(a \sqsubseteq a)$), транзитивно ($\forall a, b, c (a \sqsubseteq b) \wedge (b \sqsubseteq c) \Rightarrow a \sqsubseteq c$) и антисимметрично ($\forall a, b (a \sqsubseteq b) \wedge (b \sqsubseteq a) \Rightarrow a = b$).

Д.Скотт ввёл аксиому, утверждающую, что: “Тип данных есть частично упорядоченное множество”. Фактически можно предполагать, что всё множество элементов ограничено сверху элементом \top , т.е. $x \sqsubseteq \top$ всегда имеет место. Так же можно ввести и объект \perp , для которого всегда выполняется $\perp \sqsubseteq x$. Кроме того, можно предположить, что любые два элемента x и y имеют *наименьшую верхнюю грань*

– н.в.г. $(x \sqcup y)$ и *наибольшую нижнюю грань* – н.н.г. $(x \sqcap y)$. Пример графического отображения частично упорядоченного множества типов данных приведен на рисунке 3.

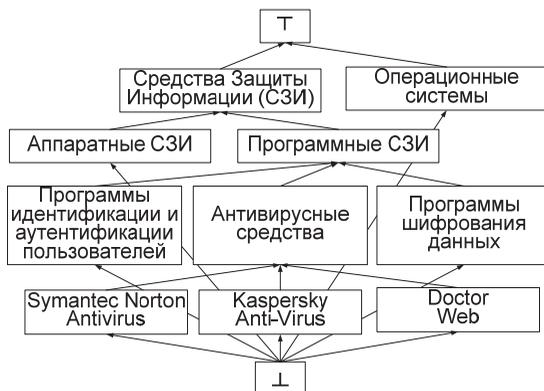


Рис. 3. Пример графического отображения частично упорядоченного множества типов данных

После того, как предположили, что тип данных допускает пределы, необходимо пересмотреть взгляд на функции. Если функция вычислима в каком-либо интуитивном смысле, то для извлечения “конечного” объема информации из какого-либо значения функции, необходимо запросить лишь “конечный” объем информации об аргументах (тут понятие информации скорее качественное, чем количественное). Однако ещё возможно выразить это фундаментальное ограничение на функции: а именно, функции должны сохранять пределы. Отображения же между типами данных *непрерывны*.

Во всей общности это можно точно выразить в терминах направленных множеств. Подмножество $X \subseteq D$ *направлено*, если каждое подмножество X имеет по крайней мере одну верхнюю грань в X . Следует заметить, что направленное множество не пусто. Функция $f: D \rightarrow D'$ непрерывна в том и только том случае, если для любого направленного множества $X \subseteq D: f(\sqcup X) = \sqcup \{f(x) : x \in X\}$.

Не все н.в.г. следует рассматривать как пределы, а только н.в.г. направленных множеств. Следует заметить, что понятие непрерывности легко распространяется на функции многих переменных. Для непрерывности функции по всем переменным одновременно достаточно требовать её непрерывности по каждой переменной в отдельности [12].

Таким образом, в более абстрактных терминах по любым двум

полным решёткам D и D' можно образовать функциональное пространство $[D \rightarrow D']$ всех непрерывных функций из первой во вторую.

Далее Д.Скотт [28] предположил, что частичный порядок является *решёткой*, а тип данных – *полной решёткой* по своему частичному порядку (это ещё одна важная аксиома, введённая Д.Скоттом).

Тип данных, удовлетворяющий двум приведённым выше аксиомам, введённым Д.Скоттом, можно рассматривать как топологическое пространство [28], а в подобных пространствах открытое множество содержит вместе с каждым элементом всё, что его аппроксимирует (см. рисунок 4).

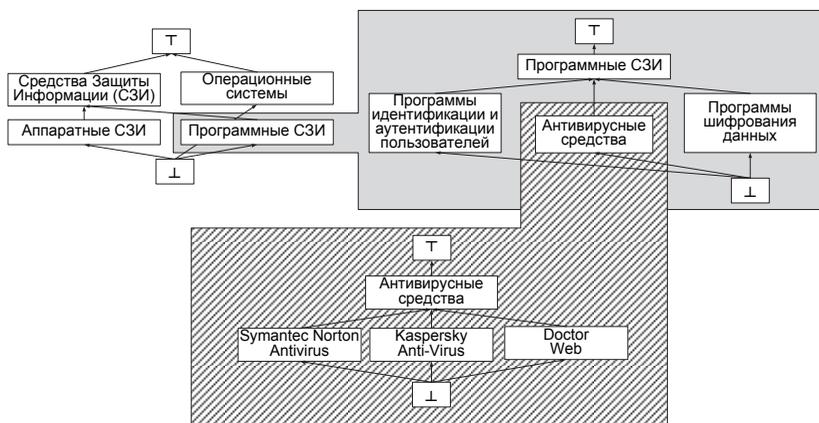


Рис. 4. Пример дуальности Концептов онтологии (Концепт может рассматриваться как объект и как тип)

Данное заключение развито в работе [13], в которой авторы предложили объединить все объекты и типы данных вместе в унифицированную область. Любой элемент полученной таким образом области выступает тогда в двух ролях:

- 1) он является *объектом*, на котором могут быть определены функции, включая функции, являющиеся наименьшими неподвижными точками рекурсивных определений;
- 2) он является *типом* всех объектов, которые аппроксимируют его; следовательно, утверждения $x \sqsubseteq y$ “ x имеет тип y ” и “любой объект типа x является объектом типа y ” эквивалентны.

По произвольной исходной области “ O ” объектов и совокупности интуитивных типов данных можно образовать новую область “ \hat{O} ” (которую называют типовым расширением области “ O ”), добавив типы как

новые объекты в смысле (1). Расширенное отношение \sqsubseteq на “ \hat{O} ” определяется эквивалентностью из (2): тип-объект помещается над объектами этого типа и над его подтипами (так что \sqsubseteq одновременно упорядочивает типы по включению и объекты по аппроксимации).

Важным понятием при описании любой ПрО, является понятие самой области. Под *областью* подразумевается [13] частично упорядоченное множество D , такое, что D имеет наименьший элемент и любое направленное множество элементов D имеет н.в.г. (такие области принято называть полными частичными порядками).

Под *типом данных* над D подразумевается подмножество x универсума D , такое, что x :

(1) замкнуто вниз, т. е. если d_0 и d_1 лежат в D и если $d_0 \sqsubseteq d_1$, то $d_1 \in x$ влечет $d_0 \in x$;

(2) замкнуто относительно взятия верхних граней, т. е. если s есть направленное подмножество x , то $\sqcup s \in x$.

Множества с этими свойствами называются также *идеалами*.

С каждым элементом d из D ассоциируется тип \tilde{d} , определяемый как: $\tilde{d} = \{d' \mid d' \sqsubseteq d\}$, т.е. \tilde{d} есть множество всех элементов, аппроксимирующих d .

Расширенная область \hat{D} образуется (фактически) добавлением к D некоторой совокупности типов.

Под *структурой типов* над D подразумевают совокупность T типов над D , удовлетворяющую следующим условиям [13]:

(1) $\tilde{d} \in T$ для всех $d \in D$;

(2) универсальный тип U (множество всех элементов из D) принадлежит T ;

(3) теоретико-множественное пересечение типов из любой непустой подколлекции, взятой из T , снова лежит в T .

Область \hat{D} – это множество T , упорядоченное по теоретико-множественному включению. Таким образом, \hat{D} содержит изоморфную копию D и поэтому может рассматриваться как расширение D .

При построении структуры типов над D может оказаться необходимым добавить в T некоторые множества с одной лишь целью – пополнить структуру так, чтобы получить необходимые пересечения (см. рисунок 5). Наличие самой структуры типов, представленной в виде решётки, позволяет говорить о возможности нахождения классов эквивалентности (о наличии в некотором роде эквивалентных типов данных, представляющих понятия из различных ПрО).

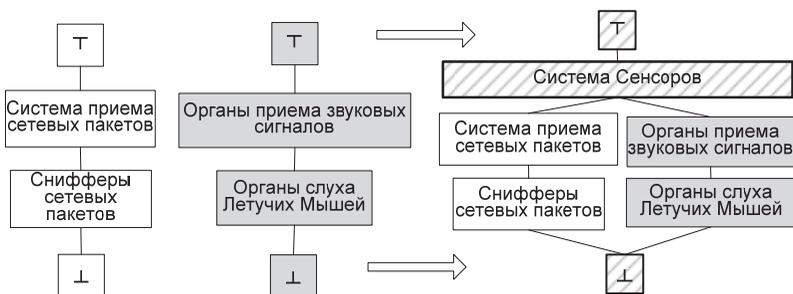


Рис. 5. Пример построения структуры типов, путём внесения множества “Система сенсоров”

При моделировании семантических вычислений весьма перспективным видится использование функциональных типов. Д.Скотт считал [29] очень важной также ту часть своих исследований, в которой он рассматривал функции, в качестве аргументов которых выступают также функции, т.е. функции высших порядков. Именно Д.Скотт одним из первых предложил в качестве семантической структуры использовать *функциональное пространство*, но многие тогда не понимали его, а ведь именно функциональные пространства должны учитываться при выборе и реализации операторов (исходя из семантики моделируемых процессов), осуществляющих отображения на функциональных пространствах. Такие операторы есть не что иное, как комбинаторы – в терминах Х.Карри и А.Чёрча (или – функциональные формы у Дж.Бэкуса).

Для упрощения восприятия и использования функциональных типов на практике, полезно иметь возможность явного задания их частично упорядоченного множества в виде Концептов соответствующего типа, а также отношений между ними (см. рисунок 6). Это должно облегчить процесс онтологического моделирования предметных областей конфликта.



Рис. 6. Пример явного задания функциональных типов и отношений между ними

Исходя из результатов проведенного анализа можно предположить, что любой тип данных, используемый при онтологическом моделировании предметных областей конфликта, можно описать через перечень (1) *объектов*, которые ему соответствуют; (2) *свойств* этих

объектов и функций (а для общего случая – (3) *действий*), которые могут рассматриваемые объекты выполнять [30].

Если рассматривать абстрактные классы объектов “*O*”, свойств “*P*” и действий “*A*”, то их в совокупности можно представить в виде решётки, содержащей один нижний, один верхний элемент и три дискретно разделённых элемента, находящихся на одной “горизонтالي”.

Следует отметить, что если учесть тот факт, что объект может состоять из объектов более низкой иерархии (более простых объектов) и быть частью объектов более высокой иерархии (структурно более сложных объектов), то в общем случае можно говорить о структурах, а не об объектах. Однако далее для простоты изложения предлагается использовать термин “объекты”, тем самым обозначая некий “срез наблюдений” в условиях определённых контекстов.

Таким образом, пусть: “*O*” – решётка “Объектов”, “*P*” – решётка “Свойств”, “*A*” – решётка “Действий”.

Исходя из анализа порядка интеллектуальной деятельности человека можно предположить, что поиск решений осуществляется в плоскости “Свойств” объектов.

При построении решётки “Свойства” предлагается воспользоваться отношением “Аппроксимирует” (“Является подклассом”).

$F1_1: [P \rightarrow P]$ Свойство аппроксимирует Свойство (Свойство является подклассом Свойства)

Пример такого построения приведён на рисунке 7.



Рис. 7. Пример отношения Аппроксимирует (“Является подклассом”)

Для возможности иерархического представления данных об объектах и их возможных действиях предлагается ввести отображения $F1_2$ и $F1_3$:

$F1_2: [O \rightarrow O]$ Объект аппроксимирует Объект

$F1_3: [A \rightarrow A]$ Действие аппроксимирует Действие

Очевидно, что об объектах судят через их свойства, которые следует рассматривать как атрибуты объектов. Следовательно, можно определить отображение $F2$:

$F2: [O \rightarrow P]$ Объект обладает Свойством

Следует заметить, что разбиение “Объектов” на подклассы может осуществляться различными способами (в зависимости от того, какое свойство выбирается за признак классификации). Возможность различной классификации свойств (а следовательно объектов и действий) порождает возможность стратификации знаний.

Реализация решений осуществляется через выполнение “Действий”, суть которых может сводиться к изменению объектов (или к измерению и/или оцениванию их свойств).

Для того, чтобы взаимодействие между двумя объектами стало потенциально возможным, они должны обладать соответствующими свойствами. Иными словами, наличие у объекта определённого свойства порождает способность к выполнению определённого действия (определённого класса действий). Само же действие, осуществляемое первым объектом, может быть пригодно для воздействия на определённое свойство второго объекта. Таким образом, можно определить ещё две функции-стрелки на решётках “Свойства” и “Действия”:

F3: $[P \rightarrow A]$ Свойство порождает способность к Действию

F4: $[A \rightarrow P]$ Действие пригодно для воздействия на Свойство

Множества данных функциональных типов (**F3** и **F4**) также образуют решётки.

Взаимодействие двух объектов (O_1 и O_2) можно формально описать следующим образом: $[[O_1 \rightarrow P_1] \rightarrow A_1] \rightarrow [O_2 \rightarrow P_2]$. Что означает: “Объект O_1 , обладающий свойством P_1 , осуществляет воздействие A_1 на объект O_2 , поскольку последний обладает свойством P_2 ”.

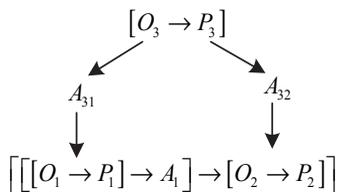
Приведённая конструкция, позволяющая строить функциональные пространства, весьма важна, так как её анализ (как инженером по знаниям, так и самой ИС) позволяет ответить на вопросы:

- “кто (что) взаимодействует” – (O_1 и O_2),
- “как взаимодействует” – (A_1),
- “почему взаимодействие возможно” – ($O_1 \rightarrow P_1$ и $O_2 \rightarrow P_2$).

Так как O_1 , O_2 , P_1 , P_2 и A_1 являются элементами соответствующих частично упорядоченных множеств (“ O ”, “ P ” и “ A ”), то это даёт предпосылки к возможности системного рассмотрения описываемого процесса “взаимодействия” и, в последующем, индуктивного и дедуктивного вывода “подобных” процессов.

В подавляющем же большинстве случаев на практике при построении онтологий используют конструкции типа: $O_1 \xrightarrow{A_1} O_2$, где O_1 и O_2 – Концепты, описывающие взаимодействующие объекты, а A_1 – Роль, носящая семантически окрашенное имя. Применение подобных конструкций не всегда приемлемо, так как в ряде случаев для интеллектуальной системы, обрабатывающей знания, представленные в онтологии, теряется семантика Роли. Если же Роль представлять конструкцией типа: $P_1 \rightarrow A_1 \rightarrow P_2$, то данное построение позволяет учитывать дополнительный контекст, извлекаемый из местоположения P_1 , P_2 и A_1 в соответствующих решётках.

В 1960-х годах Д.Маккарти [31] было предложено исчисление ситуаций, целью которого было найти способ описания результатов действий (операций) вне зависимости от проблемной области. Основной формализм ситуационного исчисления – это выражение вида $s' = result(e, s)$, где s' – ситуация, возникающая, когда происходит событие e в ситуации s . Иными словами, был обозначен вопрос, связанный с необходимостью формализовать понятие контекста. Очевидно, что проектируемая ИС просто обязана принимать те или иные решения с учётом контекста, так как именно контекст оказывает влияние на оценку значения измеряемых киберсистемой свойств. Тогда взаимодействие двух объектов в условиях определённого контекста можно в общем виде формализовано представить следующим образом:



где $[O_3 \rightarrow P_3]$ – обобщённое описание контекста (которое при необходимости можно “развернуть” в сколь угодно сложное взаимодействие объектов окружающей систему действительности), а A_{31} и A_{32} – оказываемые воздействия со стороны окружающей действительности (контекста) на первый и второй объекты соответственно.

Для того, чтобы стало возможным описывать (выводить) процессы, необходимо ввести отображение на решётке действий:

F5: $[A \rightarrow A]$ Действие следует за Действием (апликативное применение)

Бесспорно, важным элементом в познании является декомпозиция, т.о., предлагается ввести отображение на решётке объектов:

F6: $[O \rightarrow O]$ Объект состоит из Объекта

Полезно также ввести подобное отображение над решётками “Действия”, что при необходимости позволит осуществить декомпозицию рассматриваемого процесса на составляющие его действия:

F7: $[A \rightarrow A]$ Действие состоит из Действия

Основные концепты и роли (отображения между ними), используемые для построения Базы Знаний проектируемой интеллектуальной системы, можно представить в виде единой схемы: см. рисунок 8. Предложенная модель представления знаний (см. рисунок 8) может быть использована для организации семантической памяти [2] Гиromата [32], в которой будет храниться онтология, построенная им.

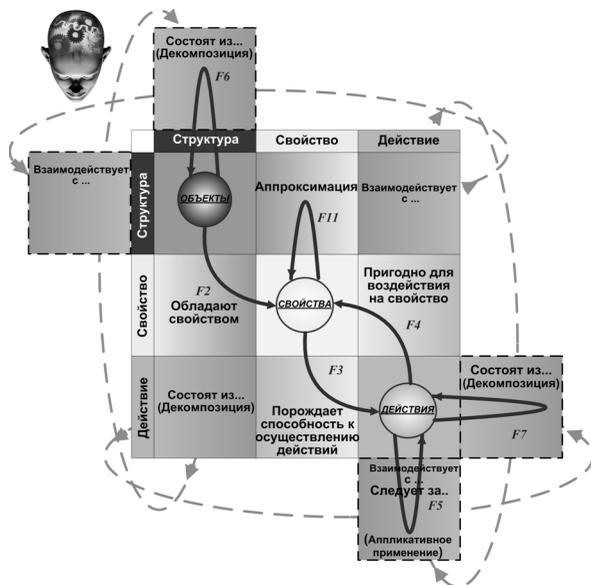


Рис. 8. Основные Концепты и Роли, которые предлагается использовать для построения Базы Знаний (семантическая память)

Перечень предложенных отображений при необходимости может быть расширен. Так, например, можно дополнительно ввести Роль между двумя объектами – “Продуцирует”. Применение указанной Ро-

ли позволит более просто описывать переход объекта из одного состояния в другое под воздействием второго объекта. Грамотное использование подобных отображений (Ролей) позволит обогатить БЗ интеллектуальной системы, что может привести к облегчению осуществления семантического вывода на имеющихся знаниях и к упрощению интерпретации полученных результатов.

Следует заметить, что предложенный порядок структурной организации данных в памяти Гиromата, соответствует порядку организации и обработки данных человеком, так как доказано, что он отдельно хранит данные (1) *о событиях и последовательностях событий* (сценариях) и (2) *об объектах и их свойствах* (каркасах). Также доказано [4], что в семантической памяти информация об объектах организована на основе различий между сенсорными или зрительными образами и функциональными свойствами, что вполне соответствует предложенному подходу по построению решёток “Объекты” и “Действия” через построение решётки “Свойства”.

4. Семантические звенья и иерархическое упорядочение концептов в онтологии. В конечном итоге знания, представленные в онтологии, предлагается хранить в виде совокупности взаимосвязанных “Семантических Звеньев” (СЗ – два Концепта, связанных Ролью), типы которых представлены выше. Однако, для того чтобы проектируемую систему можно было считать интеллектуальной, она должна быть способной не только структурировано хранить данные, но и “логически мыслить”, а также выполнять сложные когнитивные действия с целью порождения новых знаний, не закладываемых в неё непосредственно при создании. Для этого ИС должна быть способной выполнять рассуждения. При этом полезно, чтобы при необходимости эксперт, работающих с ИС, мог проверить порядок проведённых ею рассуждений.

Рассуждением называется процедура обоснования некоторого высказывания путем пошагового выведения его из других высказываний. Простейшим видом рассуждения является умозаключение (умозаключение – это непосредственный переход от одного или нескольких высказываний $A_1 A_2, \dots, A_n$ к некоторому высказыванию B). Любое высказывание, исходя из предложенной модели представления знаний [32], можно рассматривать как факт, представленный в Базе Знаний интеллектуальной системы в виде СЗ. При этом все Концепты, входящие в различные СЗ иерархически упорядочены и, как было указано выше, являются элементами частично упорядоченных множеств.

Иерархическое упорядочивание Концептов является результатом классификации, которую осуществляют при пополнении онтологии ИС. Под классификацией понимается результат последовательного деления некоторого понятия на его виды, видов на подвиды и т. д. Любая классификация может быть представлена в форме дерева понятий. “Понятия” используются при классификации объектов и явлений, ок-

ружающих ИС в ходе её функционирования. Значимость же построенных в результате классификации иерархий понятий весьма существенна как для представления знаний в памяти ИС и последующего взаимодействия с экспертами и инженерами по знаниям, так и для осуществления умозаключений самой системой. Поэтому важно указать (напомнить) лицам, заносящим в онтологию ИС новые данные, на то, каким образом следует осуществлять классификацию. В то же время следует отметить, что весьма полезно стремление к тому, чтобы ИС была способной манипулировать терминами, близкими к терминам естественного языка, так как это должно значительно облегчить процедуру человеко-машинного взаимодействия.

Термины естественного языка имеют две важнейшие характеристики: *значение* (предмет, знаком которого является термин – экстенционал) и *смысл* (информация о значении термина, которую содержит сам термин или которая связывается с ним – интенционал).

Понимать термин – значит знать, какие именно предметы попадают под него, то есть по любому предъявленному предмету уметь решить вопрос, можно ли данный предмет обозначить данным термином.

Понятие есть мысль, которая посредством указания на некоторый признак выделяет из универсума и собирает в класс (обобщает) предметы, обладающие этим признаком.

Одним из атрибутов интеллектуальной обработки информации является возможность осуществлять манипулирование не только терминами, но и пониманиями терминов.

Пусть U - универсум (*род*), тогда с синтаксической точки зрения понятия можно выражать конструкцией вида: $\alpha A(\alpha)$, которая читается как: “предмет α из универсума U , такой, что α обладает признаком $A(\alpha)$ ”. Признак $A(\alpha)$ в этом случае является *видовым отличием*. Таким образом, всякое понятие выделяет в универсуме (роде) U те и только те предметы, которые обладают видовым отличием $A(\alpha)$.

С семантической точки зрения каждое понятие обладает двумя важнейшими характеристиками – *содержанием* и *объемом*.

Содержанием понятия, выраженного универсалией $\alpha A(\alpha)$, называется признак $A(\alpha)$, на основании которого обобщаются и выделяются предметы в данном понятии.

Объемом понятия, выраженного универсалией $\alpha A(\alpha)$, называется класс всех тех предметов из универсума, которые обладают признаком $A(\alpha)$.

Объем понятия является экстенсиональной характеристикой понятия и его представляют выражением $W\alpha A(\alpha)$, которое читается: “множество тех предметов α из универсума U , для которых верно

$A(\alpha)$ ”. Те предметы, которые входят в объем понятий, называют *элементами* их объемов.

Процесс образования понятия естественно описывается в терминах гомоморфизма. Разбивая интересующее множество объектов на классы «эквивалентных» в каком-либо отношении элементов (то есть, игнорируя все различия между элементами одного класса, не интересующие исследователя в данный момент), можно получить новое множество, гомоморфное исходному (т.н. фактормножество), по выделенному отношению эквивалентности. Элементы этого нового множества (классы эквивалентности) можно воспринимать теперь как единые, нерасчленимые объекты, полученные в результате «склеивания» всех неразличимых в фиксированных отношениях исходных объектов в один «комок». Эти «комки» отождествлённых между собой образов исходных объектов и есть то, что называют понятиями, полученными в результате мысленной замены класса близких между собой представлений одним «родовым» понятием. Понятия в онтологии как раз и представляются отдельными Концептами.

Операции над понятиями бывают двух типов: операции с объемами понятий и операции с их содержаниями [33]. Взаимосвязь между объемными характеристиками понятий и их содержательными характеристиками явно отражается в *законе обратного отношения между объемами и содержаниями понятий*: “Объем понятия $\alpha A(\alpha)$ составляет часть объема понятия $\alpha B(\alpha)$, если и только если содержание $\alpha B(\alpha)$ является частью содержания $\alpha A(\alpha)$ ”.

Следует отметить, что сравнить объемы рассматриваемых понятий можно лишь при наличии некоторого дополнительного знания, описывающего область предметов U .

Учитывая указанное замечание можно более точно сформулировать закон обратного отношения, приведённый выше: $W\alpha A(\alpha) \subseteq W\alpha B(\alpha)$, если и только если $T \vdash \forall \alpha (A(\alpha) \supset B(\alpha))$ и $T \vdash \forall \alpha (B(\alpha) \supset A(\alpha))$, где левая часть записи говорит о том, что класс реально существующих предметов, образующих объем понятия $\alpha A(\alpha)$, составляет часть объема $\alpha B(\alpha)$, а правая часть записи означает, что утверждение о соответствующем отношении содержаний данных понятий имеет место в теории T (здесь под теорией следует понимать определённую “модель мира”, конкретную “предметную область”, которую ещё называют “интерпретацией”).

Над понятиями можно осуществлять целый ряд операций, необходимых для реализации “навигации” по частично упорядоченному множеству понятий (Концептов онтологии), одной из которых является *операция ограничения понятий*.

Ограничить понятие $\alpha B(\alpha)$ - это значит указать понятие $\alpha A(\alpha)$, такое что для объемов A и B будет справедливо отношение $A \subset B$. Операция ограничения $\alpha B(\alpha)$, таким образом, состоит в переходе к *видовому* понятию $\alpha A(\alpha)$. Само $\alpha B(\alpha)$ при этом считается *родовым*. Для непустых понятий пределом их ограничения считается единичное понятие.

Операцию ограничения объема понятия нельзя путать с операцией *членения* предмета. Первая является действием с понятиями, т. е. с мыслительными конструкциями, вторая же осуществляется с элементами понятий, т. е. с самими предметами.

Осуществить операцию *обобщения* понятия $\alpha A(\alpha)$ – это значит указать понятие $\alpha B(\alpha)$ такое, что будет верно отношение $A \subset B$. Процедура обобщения, таким образом, состоит в переходе от видового понятия к родовому (т.е. вверх по частично упорядоченному множеству Концептов, представленных в онтологии [32])

Пределом обобщения является универсальное понятие, т. е. некоторое понятие $\alpha D(\alpha)$ такое, что $W\alpha D(\alpha) = U$ (т.е. $\alpha D(\alpha)$ является верхом решётки).

Еще одной операцией является операция *деления* понятий. Под делением непустого понятия $\alpha B(\alpha)$ понимают построение системы понятий $S = \{\alpha A_1(\alpha), \alpha A_2(\alpha), \dots, \alpha A_n(\alpha)\}$ на основании какой-либо характеристики Ψ элементов этого понятия. При осуществлении операции деления используется следующая терминология: $\alpha B(\alpha)$ называется *делимым понятием*; понятия, входящие в систему S , называются *членами деления*, а характеристика Ψ - *основанием деления*.

Деление считается правильным, если:

1) деление осуществляется по одному основанию, т. е. в качестве Ψ используется в точности одна (простая или сложная) характеристика предметов;

2) $\forall i (W\alpha A_i(\alpha) \subset W\alpha B(\alpha))$ – члены деления, входящие в S , являются видами по отношению к понятию $\alpha B(\alpha)$;

3) $\forall i (W\alpha A_i(\alpha) \neq \emptyset)$ – члены деления не пусты;

4) $\forall i \forall j (i \neq j \supset W\alpha A_i(\alpha) \cap W\alpha A_j(\alpha) = \emptyset)$ – члены деления попарно несовместимы;

5) $W\alpha A_1(\alpha) \cup W\alpha A_2(\alpha) \cup \dots \cup W\alpha A_n(\alpha) = W\alpha B(\alpha)$ – объединение объемов всех членов деления из S совпадает с объемом $\alpha B(\alpha)$.

Невыполнение лицами, вносящими данные в ИС, какого-либо из

выше указанных требований может привести к неправильному делению и неверному составлению классификаций (построению частично упорядоченных множеств Концептов), что в свою очередь может сказаться на самой возможности “правдоподобных” умозаключений.

5. Заключение. Для того чтобы проектируемая киберсистема могла стать поистине интеллектуальной, она должна быть способной структурировано хранить данные, знания и их контекстуальную привязку к задачам и решениям, а также выполнять сложные когнитивные действия (“логически мыслить”) с целью порождения новых знаний, не закладываемых в неё непосредственно, а приобретённых ею в процессах постановки и решения возможных задач. Предложенный подход к онтологическому представлению данных в памяти интеллектуальной системы, основанный на теории типов и теории конечной аппроксимации, открывает возможность формального представления контекстно-зависимых многослойных знаний о предметных областях, потенциально содержащих знания о стратегиях поведения конфликтующих сторон. Данный подход должен позволить в дальнейшем организовать корректную процедуру осуществления правдоподобных умозаключений над контекстно-зависимыми знаниями, представленными в памяти систем предотвращения киберугроз [34].

Литература

1. *Прайд В., Медведев Д.А.* Феномен NBIC-конвергенции: Реальность и ожидания // URL: <http://www.nanonewsnet.ru/articles/2010/fenomen-nbic-konvergensii-realnost-ozhidaniya> (дата обращения: 20.12.2014).
2. *Tulving E.* Episodic and semantic memory. *Organization of Memory* // New York: Academic Press, 1972. pp. 381–403.
3. *Martin A., Chao L.L.* Semantic memory and the brain: Structure and Processes // *Current Opinion in Neurobiology*. 2001. vol. 11, pp. 194–201.
4. *Marques J.F., Canessa N., Siri S., Catricala E., Cappa S.* Conceptual knowledge in the brain: fMRI evidence for a featural organization // *Brain Research*. 2008. vol. 1194. pp. 90–99.
5. *Collins A.M., Quillian M.R.* Retrieval time from semantic memory // *Journal of Verbal Learning and Verbal Behavior*. 1969. vol. 8. pp. 240–247.
6. About formal bases of OWL // URL: http://semanticfuture.net/index.php/O_формальных_основах_OWL (дата обращения: 20.12.2014).
7. *Ontology* // URL: <http://www.aiportal.ru/articles/other/ontology.html> (дата обращения: 20.12.2014).
8. OWL, язык веб-онтологий. Краткий обзор. Рекомендация W3C от 10 февраля 2004 г. // URL: <http://www.thalion.kiev.ua/idx.php/7/009/article/#s4> (дата обращения: 20.12.2014).
9. *Norman D.A., Shallice T.* Attention to action: Willed and automatic control of behavior // *Consciousness and Self-regulation. Advances in Research and Theory*. New York: Plenum Press. 1986. vol. 4. pp. 1–18.
10. *Badddeley A.D.* The episodic buffer: A new component of working memory? // *Trends in Cognitive Sciences*, 2000. vol. 4(11). pp. 417–423.
11. *Scott D. S.* Models for various type-free calculi // *Logic, Methodology and Philosophy of Science IV (Proc. Int. Congress 1971)*, North-Holland. 1973. pp. 157–188.

12. *Scott D.S.* Outline of mathematical theory // 4th Annual Princeton Conf. on Information Sciences and Systems, Princeton University. 1970. pp. 169–176.
13. *Shamir A.* Data types as objects // Springer Berlin Heidelberg. 1977. pp. 465–479.
14. *Burstall R.* Programs and their proofs: an algebraic approach // Machine Intelligence. Edinburgh Univ. Press, 1969. no. 4.
15. *Burstall R.* The algebraic theory of recursive program schemes // Category Theory Applied to Computation and Control. 1974. no. 25. pp. 126–131.
16. *Goguen J.* Abstract data types as initial algebras and the correctness of data representation // Current Trends in Progr. Methodology, IV. Data Structuring. Prentice-Hall. 1978. pp. 80–144.
17. *Guttag J.* Abstract data types and software validation // Communications of the ACM. 1978. vol. 21(12). pp. 1048–1064.
18. *Zilles S.* Algebraic specifications for data types // IBM Research Laboratory, San Jose, California, 1975.
19. *Demers A.J.* Incapsulated Data Types and Generic Procedures // Design and Implementation of Programming Languages, LNCS 1977. no. 54. pp. 174–214.
20. *Church. A.* The calculi of lambda-conversion // Annals of Math. Studies. 1951. no. 6.
21. *Hoare C.A.R.* Notes on data structuring // Structured Programming. Academic Press. 1972. pp. 98–197.
22. *Parnas D.L.* Abstract types defined as classes of variables // Proceedings Conference on Data: Abstraction, Definition, and Structure, Salt Lake City. 1976. pp. 149–154.
23. *Kaplan M.A.* A general scheme for the automatic inference of variable types // Proceedings of the 5th ACM SIGACT-SIGPLAN symposium on Principles of programming languages. 1978. pp. 60–75.
24. *Лаэров С.С.* Основные понятия и конструкции языков программирования // М.: Статистика. 1982. 80 с.
25. *Ledgard H. F.* Ten mini-languages: a study of topical issues in programming languages // Corp. Surveys. 1971. no. 3. pp. 115–147.
26. *Lewis C.H.* Recursively defined data types // Proceedings of the 1st annual ACM SIGACT-SIGPLAN symposium on Principles of programming languages. 1973. pp. 125–138.
27. *Berry D.M.* Type equivalence in strongly typed languages: one more look // SIGPLAN Notices, 1979. no. 9. pp. 35–41.
28. *Scott D.S.* Lattice Theory. Data Types and Semantics // Formal Semantics of Programming Languages. Prentice-Hall, Englewood Cliffs, N.J. 1972.
29. *Скотт Д.С.* Логика и языки программирования // Лекции лауреатов премии Тьюринга (ред.: Эшенхерст Р.). М.: Мир, 1993. С. 65–83.
30. *Бирюков Д.Н., Ломако А.Г.* Формализация семантики для представления знаний о поведении конфликтующих сторон // Материалы 22-й научно-практической конференции “Методы и технические средства обеспечения безопасности информации”. СПб.: Издательство Политехнического университета, 2013. С. 8–11.
31. *McCarthy J., Hayes P.J.* Some philosophical problems from the standpoint of artificial intelligence // Machine Intelligence, American Elsevier, New York, 1969. no. 4.
32. *Бирюков Д. Н., Ломако А. Г.* Подход к построению ИБ-систем, способных синтезировать сценарии упрещающего поведения в информационном конфликте // INSIDE. 2014. №6. С. 42–50.
33. *Бочаров В.А., Маркин В.И.* Основы логики // М.: МГУ, 2008. 336с.
34. *Бирюков Д. Н., Ломако А. Г.* Подход к построению системы предотвращения киберугроз // Проблемы информационной безопасности. Компьютерные системы. 2013. №2. С. 13–19.

References

1. Pride V., Medvedev D.A. [NBIC convergence phenomenon: Reality and expectations]. Available at: <http://www.nanonewsnet.ru/articles/2010/fenomen-nbic-konvergensii-realnost-ozhidaniya> (accessed 20.12.2014).
2. Tulving E. Episodic and semantic memory. *Organization of Memory*. New York: Academic Press. 1972. pp. 381–403.
3. Martin A., Chao L.L. Semantic memory and the brain: Structure and Processes. *Current Opinion in Neurobiology*, 2001. vol. 11, pp. 194–201.
4. Marques J.F., Canessa N., Siri S., Catricala E., & Cappa S. Conceptual knowledge in the brain: fMRI evidence for a featural organization. *Brain Research*. 2008. vol. 1194. pp. 90–99.
5. Collins A.M., Quillian M.R. Retrieval time from semantic memory. *Journal of Verbal Learning and Verbal Behavior*. 1969. vol. 8. pp. 240–247.
6. About formal bases of OWL. Available at: <http://semanticfuture.net/index.php> (accessed 20.12.2014).
7. Ontology. Available at: <http://www.aiportal.ru/articles/other/ontology.html> (accessed 20.12.2014).
8. OWL, language of web ontologies. Short review. Recommendation of W3C. 2004. Available at: <http://www.thalion.kiev.ua/idx.php/7/009/article/#s4> (accessed 20.12.2014).
9. Norman D.A., Shallice T. Attention to action: Willed and automatic control of behavior. *Consciousness and Self-regulation. Advances in Research and Theory*. New York: Plenum Press. 1986. vol. 4. pp. 1–18.
10. Baddeley A.D. The episodic buffer: A new component of working memory?. *Trends in Cognitive Sciences*, 2000. vol. 4(11). pp. 417–423.
11. Scott D. S. Models for various type-free calculi. Logic, Methodology and Philosophy of Science IV (Proc. Int. Congress 1971), North-Holland. 1973. pp. 157–188.
12. Scott D.S. Outline of mathematical theory. 4th Annual Princeton Conf. on Information Sciences and Systems, Princeton University. 1970. pp. 169–176.
13. Shamir A. Data types as objects. Springer Berlin Heidelberg. 1977. p. 465–479.
14. Burstall R. Programs and their proofs: an algebraic approach. Machine Intelligence. Edinburgh Univ. Press, 1969. no. 4.
15. Burstall R. The algebraic theory of recursive program schemes. *Category Theory Applied to Computation and Control*. 1975. no. 25. pp. 126–131.
16. Goguen J. Abstract data types as initial algebras and the correctness of data representation. *Current Trends in Progr. Methodology, IV. Data Structuring*. Prentice-Hall. 1978. pp. 80–144.
17. Guttag J. Abstract data types and software validation. *Communications of the ACM*. 1978. vol. 21(12). pp. 1048–1064.
18. Zilles S. Algebraic specifications for data types. IBM Research Laboratory, San Jose, California, 1975.
19. Demers A.J. Incapsulated Data Types and Generic Procedures. *Design and Implementation of Programming Languages*, LNCS 1977. no. 54. pp. 174–214.
20. Church A. The calculi of lambda-conversion. *Annals of Math. Studies*. 1951. no. 6.
21. Hoare C.A.R. Notes on data structuring. *Structured Programming*. Academic Press. 1972. pp. 98–197.
22. Parnas D.L. Abstract types defined as classes of variables. *Proceedings Conference on Data: Abstraction, Definition, and Structure*, Salt Lake City. 1976. pp. 149–154.
23. Kaplan M.A. A general scheme for the automatic inference of variable types. *Proceedings of the 5th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*. 1978. pp. 60–75.
24. Lavrov S.S. *Osnovnye ponjatija i konstrukcii jazykov programmirovaniya* [Basic concepts and designs of programming languages] M.: Statistica. 1981. (In Russ).

25. Ledgard H. F. Ten mini-languages: a study of topical issues in programming languages. *Corp. Surveys*. 1971. no. 3. pp. 115–147.
26. Lewis C.H. Recursively defined data types. Proceedings of the 1st annual ACM SIGACT-SIGPLAN symposium on Principles of programming languages. 1973. pp. 125–138.
27. Berry D.M. Type equivalence in strongly typed languages: one more look. *SIGPLAN Notices*, 1979. no. 9. pp. 35–41.
28. Scott D.S. Lattice Theory. Data Types and Semantics. Formal Semantics of Programming Languages. Prentice-Hall, Englewood Cliffs, N.J. 1972.
29. Scott D.S. [Logic and programming languages]. *Lektsii laureatov premii T'yuringa – Lectures of winners of an award of Turing*. M.: Mir. 1993. pp. 65–83. (In Russ).
30. Biryukov D.N., Lomako A.G. [Formalization of semantics for representation of knowledge of behavior of conflicting parties]. *Materialy 22 nauchno-prakticheskoy konferentsiyi "Metody i tehnicheskiye sredstva obespecheniya bezopasnosti informacii"* [Materials of the 22nd scientific and practical conference "Methods and Technical Means of Safety of Information". SPB: St. Petersburg Polytechnical University]. 2013. pp. 8–11. (In Russ).
31. McCarthy J. Some philosophical problems from the standpoint of artificial. *Machine Intelligence*, American Elsevier, New York, 1969. no. 4.
32. Biryukov D.N., Lomako A.G. [Approach to creation of the IB-systems capable to synthesize scenarios of anticipatory behavior in the information conflict]. *INSIDE*. 2014. no. 6. pp. 42–50. (In Russ).
33. Bocharov V.A., Markin V.I. *Osnovy logiki* [Logic bases]. M.: MSU. 2008. 336 p.
34. Biryukov D.N., Lomako A.G. [Approach to creation of system of cyber-threats preventing]. *Problemy informatsionnoy bezopasnosti. Kompyuternie sistemy – Problems of information security. Computer systems*. SPB: St. Petersburg Polytechnical University. 2013. no. 2. pp. 13–19. (In Russ).

Бирюков Денис Николаевич — к-т техн. наук, профессор кафедры систем сбора и обработки информации Военно-космической академии имени А.Ф. Можайского. Область научных интересов: системный анализ, защита информации, интеллектуальная поддержка принятия решений. Число научных публикаций — 70. Biryukov.D.N@yandex.ru; 197198, г. Санкт-Петербург, ул. Ждановская 13.; p.т.: +7(812) 237-19-60.

Biryukov Denis Nikolaevich — Ph.D., professor of system for collecting and processing information department, Mozhaisky Military Space Academy. Research interests: system analyses, IT-Security, intelligent decision support. The number of publications — 70. Biryukov.D.N@yandex.ru; 13, Zhdanovskaya street, St.-Petersburg, 197198, Russia; office phone: +7(812) 237-19-60.

Ломako Александр Григорьевич — д-р техн. наук, профессор, профессор кафедры систем сбора и обработки информации, Военно-космическая академия имени А.Ф. Можайского. Область научных интересов: информационная безопасность, теоретическое и системное программирование, синтез и верификация корректности моделей программ. Число научных публикаций — 250. lomako_ag@mail.ru; ул. Ждановская 13, 197198, Санкт-Петербург; p.т.: +7(812) 237-19-60.

Lomako Aleksandr Grigor'evich — Ph.D., Dr. Sci., professor, professor of system for collecting and processing information department, Mozhaisky Military Space Academy. Research interests: information security, theoretical and system programming, synthesis and verification of program models. The number of publications — 250. lomako_ag@mail.ru; 13, Zhdanovskaya street, St.-Petersburg, 197198, Russia; office phone: +7(812) 237-19-60.

РЕФЕРАТ

Бирюков Д.Н., Ломако А.Г. **Денотационная семантика контекстов знаний при онтологическом моделировании предметных областей конфликта.**

Интеллектуальность проектируемых и разрабатываемых систем упреждения и пресечения атакующих воздействий (компьютерных атак) существенно зависит от их возможности осуществлять смысловую, контекстуальную и наконец, семантическую обработку данных и знаний. При этом искомая система должна иметь возможность представлять и обрабатывать знания о причинах конфликтов, процессах их протекания и следствиях конфликтов. Для этого необходимо, чтобы она обладала возможностью представлять знания об объектах (субъектах), их свойствах и процессах их взаимодействия в различных контекстах. В связи с этим, в статье предлагается подход к онтологическому представлению данных в памяти интеллектуальной системы, основанный на теории типов и теории конечной аппроксимации, который открывает возможность формального представления контекстно-зависимых многослойных знаний о поведении конфликтующих сторон. Применение предложенного подхода должно позволить моделировать семантические особенности контекстно-зависимых знаний предметных областей, учитывать их изменения и уточнения при порождении решений по предотвращению кибервоздействий на критическую информационную инфраструктуру.

SUMMARY

Biryukov D.N., Lomako A.G. **Denotational Semantics of Knowledge Contexts in Ontological Modeling of Subject Domains of the Conflict.**

Intellectuality of the designed and developed systems of anticipation and suppression of the attacking influences (computer attacks) significantly depends on their ability to carry out semantic and contextual data and knowledge processing. Thus, the required system must be able to represent and process knowledge of the causes of conflicts, processes of their course and consequences. For this purpose, it is necessary for the system to be able to represent knowledge of objects (subjects), their properties and processes of their interaction in various contexts. In this regard, the article proposes an approach to ontological data presentation in memory of an intellectual system based on the theory of types and the theory of final approximation, which opens the possibility of formal representation of context-dependent multilayered knowledge of behavior of the conflicting parties. Application of the offered approach should allow one to model semantic features of context-dependent knowledge of subject domains, to consider their changes and specifications at generation of decisions on prevention of cyber impacts on critical information infrastructure.