

ПРИМЕНЕНИЕ ЛОГИЧЕСКОЙ ТЕОРИИ ПОНЯТИЙ ПРИ РАЗРАБОТКЕ СИСТЕМ ЛОГИЧЕСКОГО УПРАВЛЕНИЯ

Выхованец В.С.

Институт проблем управления им. В.А. Трапезникова РАН, г. Москва
valery@vykhovanets.ru

Ключевые слова: логическое управление, логическая теория понятий, специализированные предметные языки, система контекстного программирования.

Введение

Проектирование систем логического управления представляет собой сложную проблему, решаемую с использованием арсенала современной дискретной математики. В этом случае автоматизацией охватывается лишь этап технического проектирования и, в гораздо меньшей степени, этап логического проектирования [1].

С другой стороны, до сих пор не разработаны языковые средства, которые позволили бы однозначно и просто понимать спецификации логического управления, изоморфно переходить от алгоритма к логическим программам, корректно вносить изменения в разработанные алгоритмы, проводить сертификацию программных и аппаратурных средств [2].

Настоящая статья посвящена методологии изучения и описания предметной области – понятийному анализу, и основанной на понятийном анализе – контекстной технологии проектирования систем логического управления. Основная цель понятийного анализа состоит в получении таких декомпозиционных схем предметной области, которые хотя и сформулированы в рамках содержательных представлений, однако обладают формальной строгостью и точностью, достаточной для прямого использования полученного высокоуровневого описания для низкоуровневой реализации дискретной обработки данных. Тем самым обеспечивается получение первичных формальных спецификаций предметной области, используемых в рамках контекстной технологии для эффективной реализации логического управления.

Суть предлагаемого подхода заключается в том, что для формальной спецификации предметной области используются две формальные системы. Первая формальная система – исчисление понятий, применена для выражения результатов понятийной декомпозиции предметной области. Вторая формальная система – специализированный предметный язык, или проблемный язык, строится для каждого класса решаемых задач и используется для описания решения.

1. Содержательная постановка задачи

Главной содержательной особенностью словесно-логической традиции в описании понятий является его понимание как некоторого мысленного «слепок» множества сущностей, обобщенно репрезентирующим его в сознании человека в виде определенно организованной совокупности существенных признаков. Последнее приводит к тому, что в рамках этой традиции сложную организацию предметного содержания понятия тщетно пытаются описать уже довольно давно. Причины неудач кроются в том, что понятие является не просто статической репрезентацией реальности, а сложно устроенным когнитивным феноменом, позволяющим изменять (перестраивать) свои собственные репрезентации в зависимости от познавательных целей субъекта [3].

Однако, внешняя репрезентация понятия в словесно-логической форме видится единственной и принципиально необходимой формой представления и обработки данных. По этой при-

чине описываемый далее проблемный подход и пополняемое множество форм многоаспектного выражения понятий является некоторым неизбежным компромиссом между понятием как когнитивным феноменом и понятием как сложно организованной совокупностью существенных признаков.

На основе выявления способов абстрагирования понятий построим понятийную структуру предметной области, где под абстракцией понимается одно из четырех отображений одних понятий в другие, которые соответствуют четырем фундаментальным способам образования понятий: обобщению, типизации, агрегации и ассоциации [4].

Выявленные в процессе анализа предметной области понятия включим в множество понятий специализированного предметного (проблемного) языка, а найденные декомпозиционные схемы предметной области преобразуем в его языковые конструкции. Языковые конструкции будем рассматривать как формы выражения понятий в тексте.

2. Логическая теория понятий

Под предметной областью будем понимать фрагмент реальной (мыслимой) действительности, представляемый некоторой совокупностью принадлежащих ему сущностей. Сущность определим как устойчивое и уникальное представление о выделенной части предметной области. Сущность воспринимается в виде своих признаков. Признак – именованная сущность, характеризующая множество своих проявлений (значений) и имеющая предметную интерпретацию (семантическую роль). Признаки, по своей сути, являются элементарными сущностями, с точностью до которых производится описание предметной области. Проблемную область определим как совокупность предметной области и решаемых в ней задач (проблем). Представления, следующие их содержательной постановки решаемых задач и определяющие один из возможных аспектов рассмотрения и декомпозиции предметной области, будем называть проблематикой.

2.1. Понятие

Понятия будем именовать, и определять схемой, интенционалом и экстенционалом. Схему понятия зададим набором признаков, на которых понятие определено. Интенционал представим набором значений взаимосвязанных признаков, позволяющим отличать сущности, принадлежащие понятию, от других сущностей. Экстенционал будем рассматривать как множество сущностей, принадлежащих понятию.

Определение 1. Формальным понятием N называется тройка

$$(1) \quad N = \begin{cases} \text{shm } N = (N_0, N_1, \dots, N_{n-1}); \\ \text{int } N = \{(V_0^j, V_1^j, \dots, V_{n-1}^j) \mid j = \overline{0, m-1}\}; \\ \text{ext } N = \{E_0, E_1, \dots, E_{u-1}\}, \end{cases}$$

состоящая из схемы $\text{shm } N$, интенционала $\text{int } N$ и экстенционала $\text{ext } N$, где N – имя определяемого понятия, N_i – понятия-признаки ($i = \overline{0, n-1}$), $V_0^j, V_1^j, \dots, V_{n-1}^j$ – наборы значений понятий-признаков, составляющие интенционал определяемого понятия ($j = \overline{0, m-1}$), E_k – сущности, принадлежащие понятию N ($k = \overline{0, u-1}$).

Если некоторые исходные понятия N_i служат признаками для определения другого понятия N , то упорядоченные наборы $(V_0^j, V_1^j, \dots, V_{n-1}^j)$ следует интерпретировать как состоящие из сущностей исходных понятий, представленных в одной из возможных форм своего выражения.

2.2. Понятийная структура

В отличие от известных формализмов, где на понятиях задаются отношения различной природы, понятийную структуру опишем множеством понятий с четырьмя видами отображе-

ний, единственное назначение которых – показать способы образования понятий, или способы их абстрагирования (типизацию, агрегацию, обобщение, ассоциацию).

Абстрагирование – форма мышления, при которой происходит образование (выявление, идентификация) понятия. При абстрагировании между понятиями выявляется независимость, дифференциация и интеграция признаков. Понятия независимы, если их признаки не пересекаются. Если у двух понятий имеются общие признаки, то наблюдается дифференциация понятий. Если все признаки одного понятия являются признаками другого понятия, то происходит их интеграция.

Известны следующие абстракции [6]: обобщение (специализация), типизация (конкретизация), агрегация (декомпозиция) и ассоциация (индивидуализация). Обобщение и типизация, и обратные им специализация и конкретизация, выражают общность понятий, проявляющуюся при дифференциации. Агрегация и ассоциация, и обратные им декомпозиция и индивидуализация, раскрывают интеграцию понятий.

При обобщении происходит порождение нового понятия на основе одного или нескольких подобных понятий, когда порождаемое понятие сохраняет общие признаки исходных понятий, и игнорирует их различия. Обобщение – порождение понятия на основе пересечения схем обобщаемых понятий и расширенного объединения их экстенционалов. При специализации, наоборот, из понятия-обобщения восстанавливают одно из обобщенных в нем понятий.

Типизация является частным случаем обобщения. В отличие от обобщения при типизации имеется возможность для каждой сущности из экстенционала понятия-типа узнать ее исходное понятие. Для этого используется множество признаков, называемое ключом. Таким образом, типизация – порождение понятия на основе пересечения схем типизируемых понятий и объединения их экстенционалов. При конкретизации понятия-типа фиксируется одно из типизированных в нем понятий, для чего необходим ключ.

При ассоциации устанавливается взаимосвязь между сущностями одного и того же или разных понятий. Ассоциация выражает специфическое соединение сущностей. Это соединение позволяет от сущности одного понятия перейти к одной или нескольким сущностям других понятий. Ассоциация – порождение понятия на основе объединения схем ассоциируемых понятий и ограниченного декартового произведения их экстенционалов. При индивидуализации из понятия-ассоциации выделяются ассоциированные в нем понятия. Для перехода между сущностями этих понятий используется набор признаков, называемый связью.

При агрегации понятие строится как совокупность других понятий. Процесс, противоположный агрегации называется декомпозицией. Агрегация – порождение понятия на основе объединения схем агрегируемых понятий и декартового произведения их экстенционалов. При декомпозиции понятие-агрегат разделяется на входящие в него агрегированные понятия.

Агрегация является предельным случаем ассоциации. В отличие от ассоциации, где между сущностями устанавливаются только часть связей, при агрегации присутствуют все возможные связи: на одном и том же множестве понятий можно задать несколько ассоциаций, в то время как их агрегация единственна.

Определение 2. Под понятийной структурой будем понимать совокупность понятий, для которых заданы способы их образования (абстрагирования). Носителем понятийной структуры является множество понятий, а ее сигнатурой – множество отображений обобщения, типизации, агрегации и ассоциации.

2.3. Синтаксическая теория

Определение 3. Формулой формальной теории понятий будем называть понятийную структуру, которая дополнена описанием интенционалов всех входящих в нее понятий. Формула формальной теории понятий является:

- синтаксически полной, если в ней отсутствуют понятия, которые не содержат ни одной сущности, т.е. экстенционал которых пуст;
- семантически полной, если в ней определены все понятия, способы их абстрагирования и формы выражения, необходимые для описания заданной области интерпретации;
- противоречивой, если в ней описана сущность, выраженная как принадлежащая, так и как не принадлежащая экстенционалу одного и того же понятия.

Определение 4. Формальную теорию понятий будем называть:

- непротиворечивой, если все выводимые в теории формулы непротиворечивы;
- семантически полной, если для любой области интерпретации существует вывод соответствующей ей формулы;
- синтаксически полной, если для любой выводимой формулы существует область интерпретации;
- разрешимой, если существуют конструктивные средства для распознавания формул, выводимых в теории.

2.4. Исчисление понятий

Синтаксическую теорию понятий будем строить на основе соответствующего исчисления. Исчисление понятий определим как исчисление понятийных структур, а саму понятийную структуру будем задавать множеством формул, каждая из которых выражает образование одного из ее понятий [7]. Универсумом исчисления понятий является множество понятийных структур, а ее предмет – понятия и отображения абстрагирования.

Алфавит исчисления включает знаки понятий, знак отсутствия определения понятия \neg , знаки операций объединения, пересечения и включения множеств, знаки круглых скобок. Исчисления понятий строятся на основе теории множеств с одной дополнительной аксиомой – аксиомой существования пустого понятия:

$$(2) \frac{()()}{()},$$

где использована следующая нотация: в числителе задается имя понятия и способы его абстрагирования (слева – список обобщения, справа – список ассоциации), а в знаменателе – схема понятия. Для порождения формул исчисления используются четыре правила вывода:

$$(3) \frac{()()}{()} \stackrel{\neg N}{\mapsto} \frac{()N()}{(N)};$$

$$(4) \frac{(\dots)N_1(\dots) \dots (\dots)N_m(\dots)}{\text{shm } N_1 \dots \text{shm } N_m} \stackrel{\neg N}{\mapsto} \frac{(N_1 \dots N_m)N()}{\bigcap_{i=1}^m \text{shm } N_i \supset \{ \} \quad \bigcap_{i=1}^m \text{shm } N_i};$$

$$(5) \frac{(\dots)N_1(\dots) \dots (\dots)N_m(\dots)}{\text{shm } N_1 \dots \text{shm } N_m} \stackrel{\neg N}{\mapsto} \frac{()N(N_1 \dots N_m)}{\bigcup_{i=1}^m \text{shm } N_i \supset \{ \} \quad \bigcup_{i=1}^m \text{shm } N_i};$$

$$(6) \frac{(\dots)N_1(\dots) \dots (\dots)N_m(\dots)}{\text{shm } N_1 \dots \text{shm } N_m} \stackrel{\neg N}{\mapsto} \frac{(N_1 \dots N_t)N(N_{t+1} \dots N_m)}{\bigcap_{i=1}^t \text{shm } N_i \supseteq \bigcup_{j=t+1}^m \text{shm } N_j \quad \bigcup_{j=t+1}^m \text{shm } N_j},$$

где в левой части правил (до знака \mapsto) задаются посылки, а в правой части – заключение; над и под знаком вывода указаны условия применения правил.

Аксиома (2) утверждает, что существует как минимум одно понятие, называемое пустым. Правило (2) используется для порождения простых понятий (понятий-признаков). Правила (4) и (5) предназначены для образования составных понятий на основе дифференциации и инте-

грации признаков. Правило (6) служит для выражения сложных понятий при одновременной интеграции и дифференциации признаков других понятий.

Из определения понятия следует, что каждое понятие обладает схемой – набором признаков, или простых понятий, на которых понятие определено. Однако, при задании понятийной структуры определению подлежат только отображения одних понятий в другие. В связи с чем возникает задача определения схем понятий по понятийной структуре. Схему произвольного понятия будем вычислять на основе отображений абстрагирования. Для этого воспользуемся следующей рекуррентной процедурой:

- схема простого понятия N равна (N) ;
- схема понятия, полученного в результате дифференциации, равна пересечению схем дифференцируемых понятий;
- схема понятия, полученного в результате интеграции, равна объединению схем интегрируемых понятий;
- схема понятия, полученного в результате дифференциации и интеграции, равна объединению схем интегрируемых понятий, принадлежащему пересечению схем дифференцируемых понятий.

Операции объединения и пересечения множеств в рассматриваемом случае выполняются с учетом повторения элементов. Требование вычислимости схем понятий является распространением свойства регулярности (фундированности) множеств на понятия. В этом случае вычислимость схем понятий по понятийной структуре гарантирует отсутствие определений понятий через самих себя, что в любой формальной или содержательной теории, претендующей на адекватность, признается недопустимым.

Правила вывода исчисления понятий сохраняют синтаксическую полноту и непротиворечивость каждой выводимой в исчислении понятийной структуры. При этом аксиома рассматривается как минимальная понятийная структура, являющаяся синтаксически полной и непротиворечивой по определению. Следовательно, каждая выведенная в исчислении понятийная структура является ее теоремой. Более того, имеется возможность установить принадлежность произвольной строки, выводимой в исчислении, множеству формул (понятийных структур). Отсюда может быть показана справедливость следующего утверждения.

Утверждение 1. Исчисление понятий синтаксически и семантически полно, непротиворечиво и разрешимо.

2.5. Логическая теория

Логическую теорию понятий будем рассматривать как расширение исчисления понятий за счет дополнения его формализма исчислением предикатов первого порядка, которое применим для выражения интенционалов. В логической нотации [5] интенционал в (1) выражается формулой исчисления предикатов $f(E, V_i, N_j)$, в которую входит свободная предметная переменная E , обозначающая сущность, проверяемую на принадлежность понятию, предметные константы V_i , интерпретируемые как сущности-признаки или их множества, и предикаты N_j , выражающие наличие у сущности значений соответствующих признаков. Интенционал простого понятия-признака N представляется логическим выражением

$$N(E, \text{dom } N) \rightarrow N(E),$$

или, если домен $\text{dom } N$ признака N задан в виде нескольких множеств V_i ,

$$\bigvee_{i=0}^{n-1} N(E, V_i) \rightarrow N(E),$$

где $N(E, V_i)$ – истинный предикат, если сущность E принадлежит множеству V_i , и ложный – в противном случае.

Составные понятия описываются логическим выражением общего вида:

$$(7) \quad \bigvee_{j=0}^{m-1} \bigwedge_{i=0}^{n-1} N_i(E, V_{ij}) = \bigvee_{j=0}^{m-1} N_j(E) \rightarrow N(E),$$

где N_i и V_{ij} – понятия-признаки и заданные для них множества сущностей, а $N_i(E, V_{ij})$ – предикат принадлежности признака N_i понятия-сущности E множеству V_{ij} .

Фактически выражение (7) определяет понятие N через m альтернативных понятий N_j , которые заданы на общем множестве из n признаков и выражаются конъюнкциями своих предикатов. Значения признаков V_{ij} могут принимать произвольные множества значений из доменов $\text{dom } N_i$, соответствующих альтернативным понятиям-признакам N_j .

Для синтаксической полноты понятийной структуры требуется не пустота схемы и существования хотя бы одной сущности, для которой интенционал выполним. Это равносильно выводимости интенционала понятий в исчислении предикатов и его выполнимость при каких-либо значениях предикатных переменных, т.е. формула интенционала должна быть не тождественно-ложной. Заметим, что определение тождественной ложности формул в исчислении высказываний разрешимо. Однако проблема разрешимости исчисления предикатов решается отрицательно [8, с. 157]. По этой причине логическая теория понятий будет непротиворечивой и синтаксической неполнотой. Но имеется один частный случай, относительно которого исчисление предикатов разрешимо – это одноместное исчисление [9, с. 250]. Следовательно, только одноместное исчисление предикатов может служить формальным средством для синтаксически полного выражения интенционалов понятий.

Утверждение 2. Логическая теория понятий является разрешимой, непротиворечивой и синтаксически полной относительно одноместного исчисления предикатов.

Таким образом, для обеспечения синтаксической полноты логической теории дополним правила вывода (3-6) еще одним условием, которое устанавливает отсутствие противоречия при выражении интенционала каждого понятия.

Однако, открытым остался вопрос о семантической полноте логической теории, который может быть переформулирован следующим образом: существуют ли экстенционалы понятий произвольной предметной области, которые не выражаются на языке одноместного исчисления предикатов. Ответ на этот вопрос оказывается отрицательным даже для полного исчисления предикатов, несмотря на то, что это исчисление является непротиворечивой и семантически полной формальной теорией [8, с. 163]. Перенос этого результата в логическую теорию понятий означает, что существуют понятия, экстенционалы которых не могут быть выражены на языке исчисления предикатов. Следовательно семантическую полноту логической теории следует рассматривать относительно не произвольных областей интерпретации, а относительно некоторого их подмножества.

Определение 5. Областью логической интерпретации называется такая область интерпретации, в которой экстенционал произвольного понятия выразим на языке одноместного исчисления предикатов.

После такого определения справедливым будет следующее утверждение.

Утверждение 3. Логическая теория понятий является семантически полной относительно области логической интерпретации.

Заметим, что логическая теория понятий, хотя и является семантически полной в области логической интерпретации, однако может породить семантически неполные понятийные структуры (формулы). Это связано с тем, что, как и в исчислении предикатов, работающем с одним понятием – понятием логической истины, которое присуще всем «мыслимым мирам», так и в синтаксической теории понятий, которая работает со всеми мыслимыми понятиями, требуется индивидуальное доказательство интерпретируемости каждой выведенной формулы в заданной предметной области. Такое доказательство не может быть получено средствами синтаксической

теории, так как для его осуществления необходимо привлечение содержательных представлений относительно конкретной области интерпретации.

3. Разработка систем логического управления

Рассмотрим создание систем логического управления в рамках контекстной технологии моделирования, которая является одной из реализаций логической теории понятий и в которой в качестве программы используется описание предметной области, дополненное ситуационным описанием, выражающим решение прикладной задачи на определенном в процессе моделирования проблемном языке [11].

3.1. Понятийный анализ

Понятия, выявленные в процессе анализа предметной области, условно разделим на две группы: терминальные, или сигнификативные, выражаемые последовательностью знаков терминального алфавита проблемного языка, и нетерминальные, или денотационные, соответствующие нетерминальным знакам порождающей грамматики этого языка. Разделение понятий на денотационные и сигнификативные осуществим с учетом некоторой фиксированной проблематики, задающей класс задач, для которых определяется проблемный язык.

Понятийный анализ выполняется с учетом некоторой активной проблематики и состоит из следующих этапов:

- разделение сущностей предметной области на сигнификативные и денотационные;
- означивание сигнификативных сущностей и выявление существенных признаков у денотационных;
- сопоставление денотационных сущностей и определение их общих и различающихся признаков;
- образование новых или определение уже существующих понятий на основе интеграции и дифференциации признаков;
- создание понятийной структуры предметной области путем описания отображений одних понятий на другие;
- уточнение способа абстрагирования понятий (обобщение или типизация, ассоциация или агрегация);
- вычисление схем понятий и задание ключей – для типизации, связей – для ассоциации;
- верификации понятийной структуры путем проверки ее на полноту и непротиворечивость.

3.2. Протоязык

Протоязык, или первичный язык, предназначен для описания синтаксиса проблемных языков и реализует аксиому, которая необходима для определения базовых семантических категорий, используемых при индуктивном описании семантики проблемных языков. На рис. 1 приведена формальная грамматика протоязыка, где нетерминальные знаки грамматики обозначены строками над терминальным алфавитом, например, *cognition*, *essences* в строке 1, а терминальные знаки заключены в одинарные кавычки, например ' в строке 2.

В квадратных скобках указаны части продукций, которые могут быть опущены, например [situation] в строке 1. Альтернативные правые части правил грамматики разделены вертикальной чертой, например *pattern* | *term* в строке 8. Грамматика задана с точностью до пробелов и обозначенных курсивом:

- нетерминальных знаков *notion*, служащих для выражения определяемых нетерминальных знаков проблемных языков (строка 2);
- терминальных знаков *terminal*, образующими допустимые языковые выражения проблемного языка *text* (строка 11);

- синонимов понятий *alias*, предназначенных для ссылок на понятия абстракций и определяемых языковых конструкций (строки 3, 7);
- имен аспектов *aspect*, используемых для задания различных прагматик при интерпретации проблемного языка (строка 14).

1	cognition	→	essences [situation] [cognition]
2	essences	→	'(' [notions] ')' <i>notion</i> '(' [notions] ')' [intension]
3	notions	→	notion [<i>alias</i>] notion [<i>alias</i>] [notions]
4	intension	→	sentence sentence [intension]
5	sentence	→	syntax semantics
6	syntax	→	item [axiom] '(' syntax ')' syntax syntax syntax '(' syntax
7	item	→	notion [<i>alias</i>] lexeme
8	lexeme	→	pattern term
9	pattern	→	"" [text] ""
10	term	→	" [text] "
11	text	→	<i>terminal</i> <i>terminal</i> [text]
12	axiom	→	'[[text]]'
13	semantics	→	pragmatic [semantics]
14	pragmatic	→	[<i>aspect</i>] '{' [text] '}'
15	situation	→	[<i>aspect</i>] '<' [text] '>'

Рис. 1. Грамматика протоязыка

3.3. Онтологическое описание проблемной области

Средствами протоязыка определяется проблемный язык, на котором в ситуационной части *situation* описывается решение прикладной задачи. Для этого знания *cognition* относительно заданной предметной области представляются как состоящие из описаний ее сущностей *essences* (строка 1).

Сущностям присваивается имя *notion* нетерминального понятия определяемого проблемного языка, а само понятие задается как дифференциация (первые круглые скобки) и интеграция (вторые круглые скобки) ранее определенных понятий *notions* (строки 2, 3).

Содержание *intension* нетерминального понятия *notion* состоит из предложений *sentence*, которые служат для выражения синтаксиса *syntax* предложений, выражающих это понятие в тексте (строка 4).

3.4. Синтаксис проблемного языка

Синтаксис предложения *syntax* задается логическим выражением со скобками, состоящим из связок «и» (пусто) и «или» ('|') над элементами *item*: ранее объявленными понятиями *notion* и лексемами *lexeme* (строки 6, 7). Лексема является терминальным понятием определяемого языка. Для выражения лексем используются как терминальная строка *term*, так и множества терминальных строк, задаваемых на языке регулярных выражений в виде шаблонов *pattern* (строки 8-10).

3.5. Семантика проблемного языка

После каждого предложения *sentence* описывается его семантика *semantics* (строка 5), выражаемая множеством прагматик (строка 13). Каждая прагматика имеет имя *aspect* и выражается в виде текста, построенного по правилам определяемого проблемного языка (строка 14). Результат компиляции текста *text*, заключенного в фигурные скобки, оформляется в виде единицы вызова (императива), используемого интерпретатора (процессора, виртуальной машины,

другой системы программирования и т.п.). Если в тексте `text` будет распознано применение предложения `sentence` в контексте `aspect`, то в область кода записывается вызов императива этого предложения с именем `aspect`.

Для описания семантики первичных семантических категорий проблемного языка используется аксиома `axiom` (строка 6). Если аксиома не содержит текста `text`, то в область кода записывается значение, определяемое предыдущим элементом `item`. Если в квадратных скобках находится текст `text`, выраженный на проблемном языке, то в область кода копируется результат его компиляции.

В итоге получаем, что семантика проблемного языка задается на основе метода математической индукции путем декларации первичных семантических категорий, непосредственно реализуемых интерпретатором, и описания новых семантических категорий текстами на проблемном языке, для чего могут использоваться все ранее определенные семантические категории.

3.6. Демонстрационный пример

Рассмотрим реализацию логического управления на примере устройства, состоящем из двух клапанов с памятью и двух кнопок без памяти [10, с. 544]. Необходимо реализовать устройство управления, которое при нажатии кнопки `X1` подает управляющий сигнал на открытие клапана `K1`. После его открытия по сигналу открытого положения `X4` снимается управляющий сигнал с клапана `K1` и начинает открываться клапан `K2`. После его открытия по сигналу открытого положения `X6` снимается управляющий сигнал с клапана `K2`, после чего устройство переходит в устойчивое состояние с открытыми клапанами. В свою очередь, при нажатии кнопки `X2` подает управляющий сигнал на закрытие клапана `K2`. После его закрытия по сигналу закрытого положения `X5` снимается управляющий сигнал с клапана `K2` и начинает закрываться клапан `K1`. После его закрытия по сигналу закрытого положения `X3` снимается управляющий сигнал с клапана `K1`, после чего устройство вновь переходит в устойчивое состояние с закрытыми клапанами.

В результате понятийного анализа может быть установлено, что денотационными сущностями, используемыми при описании задачи, являются сущности понятий `Сигнал` и `Команда`. Остальные сущности являются одиночными и могут быть выражены сигнификативным способом. Далее определим проблемный язык, необходимый для описания работы устройства и соответствующий имеющемуся описанию на естественном языке (рис. 2).

В строках 1, 3 и 6 описана понятийная структура проблемной области, состоящая из трех не связанных между собой понятий: пустое понятие, `Сигнал` и `Команда`. Для реализации логического управления выберем, например, микроконтроллер на базе процессора `Intel 80386`. Для порождения исполняемого кода воспользуемся внешней системой программирования – ассемблером целевой вычислительной платформы. Для передачи ассемблеру строк исходного кода (последовательности знаков в апострофах) определим специальное предложение, выражающее пустое понятие (строка 2).

Понятие `Сигнал` содержит шесть сущностей, которые опишем отдельными предложениями. Семантику этих предложений зададим через аксиому в виде команд целевой платформы, которые заносят в стек адреса соответствующих кнопок и датчиков в адресном пространстве процессора. Аналогичным образом опишем сущности понятия `Команда`.

Так как технические устройства достаточно инерционны, определим выразительные средства для реализации задержек (строка 10). Задержка, равная времени 100 повторений двух команд процесса выражается запятой, а в два раза большая – точкой. Для задания семантики предложения первой задержки использована аксиома, а семантика второй задержки выражена на проблемном языке в виде текста из двух запятых.

```

1  () ()
2  "" "[^]+" [] "" {}
3  () Сигнал ()
4  'X1' ['push 0201h;']{} 'X2' ['push 0202h;']{} 'X3' ['push 0203h;']{}
5  'X4' ['push 0204h;']{} 'X5' ['push 0205h;']{} 'X6' ['push 0206h;']{}
6  () Команда ()
7  'Открытие K1' ['push 0301h;']{} 'Закрытие K1' ['push 0302h;']{}
8  'Открытие K2' ['push 0303h;']{} 'Закрытие K2' ['push 0304h;']{}
9  () ()
10 ' ' {mov eax, 100; L1: inc eax; jnz L1;} ' ' {, , }
11 'Выдать' Команда {pop ebx; mov [ebx], 1;}
12 'Снять' Команда {pop ebx; mov [ebx], 0;}
13 'Ожидать' Сигнал {pop ebx; L2: mov eax, [ebx]; test eax, 0; je L2;}
14 'Начало' ['L3: lea eax, L3; push eax;'] "Повторить" ['pop eax; jmp eax;'] {}

```

Рис. 2. Проблемный язык

Реализация предложений проблемного языка для выдачи и снятия команд, а также ожидания срабатывания датчиков и нажатия кнопок, тривиальна: из стека извлекается адрес порта команды (датчика, кнопки), после чего выполняются необходимые действия (строки 11-13). Для циклического выполнения команд, порожденных описанием функционирования управляющего устройства, предусмотрено предложение «Начало "Повторить"», в котором пустое понятие обозначено двойными одинарными кавычками.

Заключительная часть решения задачи – перевод описания работы системы логического управления с естественного языка на созданный проблемный язык (рис. 3).

```

< Начало.
Ожидать X1, Выдать Открытие K1.
Ожидать X4, Снять Открытие K1, Выдать Открытие K2.
Ожидать X6, Снять Открытие K2.
Ожидать X2, Выдать Закрытие K2.
Ожидать X5, Снять Закрытие K2, Выдать Закрытие K1.
Ожидать X3, Снять Закрытие K1.
Повторить. >

```

Рис. 3. Ситуационное описание

Конечно, можно использовать общую теорию автоматов, принудительное кодирование состояний, процедуры определения полноты и непротиворечивости автоматных моделей, сложные инструментальные средства программирования для промышленных контроллеров, и др. Однако, зачастую создание частной теории в виде проблемного языка, которая минимально достаточна для выражения знаний эксперта, требует значительно меньших затрат, чем решение частной задачи общими средствами. В этом случае достигаются такие преимущества как простота верификации программы управления, легкость внесения изменений, компактность описания, самодокументированность, и др.

В связи с этим укажем на то, что имеющееся исходное описание работы устройства на естественном языке занимает 709 знаков, а полученная нами реализация логического управления выражена 726 знаками (551 знак – описание протоязыка, 175 знаков – ситуационное описание),

причем в исходном тексте отсутствует определение синтаксиса и семантики используемого языка, в то время как разработанный проблемный язык задан полностью, точно и однозначно.

Заключение

Логическая теория понятий позволяет полно и непротиворечиво описать предметные области, у которых разрешающая процедура для экстенционалов понятий задается их интенционалами, выраженными формулами одноместного исчисления предикатов. В отличие от известных языков логического программирования, где стараются избегать явных онтологических допущений и произвольных форм выражения прикладных понятий, разработанная технология создания систем логического управления содержит выразительные средства, которые побуждают разработчика так специфицировать предметную область и описывать решение задач, чтобы последние имели однозначное и простое выражение, позволяли корректно вносить изменения в разрабатываемые программы, были легко верифицируемы.

В заключении укажем, что для повышения эффективности системы контекстного программирования разработан метод разнесенного грамматического разбора [12], заключающийся в разделении определения применимости конструкций проблемного языка на две части: контекстное сопоставление, осуществляемое при просмотре текста назад, и структурное распознавание, выполняемое при просмотре вперед.

Литература

1. *Закревский А. Д.* Параллельные алгоритмы логического управления. Минск, 2003. 200 с.
2. *Шальто А. А.* Алгоритмизация и программирование для систем логического управления и «реактивных» систем // Автоматика и телемеханика. 2001, № 1. С.3-39.
3. *Кравцов Л. Г.* Методологические проблемы психологического анализа мышления в понятиях // Материалы Первой российской конференции по когнитивной науке. Казань, Казанский гос. ун-т, 2004 (<http://www.ksu.ru/ss/cogsci04/>).
4. *Выхованец В. С.* Что истинно во всех мирах // Матер. межд. науч.-практ. конф. «Математическое моделирование в образовании, науке и производстве». Тирасполь, 2007. С. 36.
5. *Гаскаров Д. В.* Интеллектуальные информационные системы. М.: Высш. шк., 2003.
6. *Brodie M. L., Mylopoulos J., Schmidt J. W.* On Conceptual Modeling. New York: Springer-Verlag, 1984.
7. *Выхованец В. С.* Исчисление понятий // Тез. докл. межд. конф. «Когнитивный анализ и управление развитием ситуаций». М., 2007. С. 87-88.
8. *Эдельман С. Л.* Математическая логика. М., Наука, 1975.
9. *Гильберт Д., Бернайс П.* Основания математики. Логические исчисления и формализация арифметики. М.: Наука, 1979.
10. *Шальто А. А.* SWITCH-технология. Алгоритмизация и программирование задач логического управления. СПб.: Наука, 1998.
11. *Выхованец В. С., Иосенкин В. Я.* Понятийный анализ и контекстная технология программирования // Проблемы управления. 2005. № 4. С. 2-11.
12. *Выхованец В. С.* Разнесенный грамматический разбор // Проблемы управления. 2006. № 3. С. 32-43.