ФЕДЕРАЛЬНОЕ АГЕНТСТВО
ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО 18629-11— 2010

Системы промышленной автоматизации и интеграция

ЯЗЫК СПЕЦИФИКАЦИЙ ПРОЦЕССА

Часть 11

Ядро PSL

ISO 18629-11:2005 Industrial automation systems and integration — Process specification language — Part 11: PSL core (IDT)

Издание официальное

Москва
Стандартинформ
2014

gost.e-ecolog.ru

Предисловие

1 ПОДГОТОВЛЕН Научно-техническим центром «ИНТЕК» на основе собственного аутентичного перевода на русский язык стандарта, указанного в пункте 4

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 100 «Стратегический и инновационный менеджмент»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 21 декабря 2010 г. № 893-ст

4 Настоящий стандарт идентичен международному стандарту ИСО 18629-11:2005 «Системы промышленной автоматизации и интеграция. Язык спецификаций процесса. Часть 11. Ядро PSL» (ISO 18629-11:2005 «Industrial automation systems and integration — Process specification language — Part 11: PSL core»).

При применении настоящего стандарта рекомендуется использовать вместо ссылочных международных стандартов соответствующие им национальные стандарты Российской Федерации, сведения о которых приведены в дополнительном приложении ДА

4 ВВЕДЕН ВПЕРВЫЕ

Правила применения настоящего стандарта установлены в ГОСТ Р 1.0—2012 (раздел 8). Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (gost.ru)

© Стандартинформ, 2014

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

Содержание

1 Область применения . . . 1
2 Нормативные ссылки . . . 1
3 Термины, определения и сокращения . . . 1
3.1 Термины и определения . . . 1
3.2 Сокращения . . . 4
4 Общие положения комплекса стандартов ИСО 18629 . . . 4
5 Синтаксис спецификации языка . . . 5
5.1 Основные символы и синтаксические категории . . . 5
5.2 Лексика . . . 5
5.3 Грамматика . . . 6
5.4 Язык . . . 6
6 Основные элементы ядра PSL . . . 6
6.1 Основные свойства . . . 6
6.2 Элементарный словарь ядра PSL . . . 6
6.3 Определенная лексика ядра PSL . . . 9
6.4 Аксиомы . . . 10
7 Соответствие ядру PSL . . . 12
7.1 Соответствие онтологий . . . 12
7.2 Соответствие описаний процесса . . . 12
Приложение А (обязательное) Идентификатор ASN.1, присвоенный ИСО18629-11 . . . 13
Приложение В (справочное) Синтаксис формата обмена знаниями и семантика . . . 14
Приложение С (справочное) Пример описания процесса с использованием ядра PSL . . . 19
Приложение D (справочное) Условные обозначения формы Бэкуса-Наура . . . 25
Приложение ДА (справочное) Сведения о соответствии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации . . . 26
Библиография . . . 27

Введение

Комплекс стандартов ИСО 18629 устанавливает требования к интерпретируемому компьютером обмену информацией о производственном процессе. Стандарты комплекса определяют родовой язык, используемый для описания производственного процесса, осуществляемого одной и той же промышленной компанией или несколькими ее промышленными секторами, или разными компаниями, независимо от любой частной модели представления. Характер этого языка подходит для совместного использования информации, относящейся к производству, на всех стадиях производственного процесса.

В настоящем стандарте приведено описание базовых элементов языка, определенного в комплексе стандартов ИСО 18629.

Стандарты комплекса ИСО 18629 не зависят от какого-либо представления специфического процесса или модели, предложенной в приложении программного обеспечения в домене управления производством. Комплекс стандартов ИСО 18629 определяет структурную основу, обеспечивающую возможность взаимодействия приложений программного обеспечения.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Системы промышленной автоматизации и интеграция

ЯЗЫК СПЕЦИФИКАЦИЙ ПРОЦЕССА

Часть 11

Ядро PSL

Industrial automation systems and integration. Process specification language.
Part 11. PSL core

Дата введения — 2011—09—01

1 Область применения

Настоящий стандарт распространяется на концепцию ядра PSL, определяемую с помощью аксиом, написанных на языке, соответствующем требованиям комплекса стандартов ИСО 18629.

Концепция, приведенная в настоящем стандарте, является общей для всех процессов производства.

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие стандарты, которые необходимо учитывать при использовании настоящего стандарта. В случае ссылок на документы, у которых указана дата утверждения, необходимо пользоваться только указанной редакцией. В случае, когда дата утверждения не приведена, следует пользоваться последней редакцией ссылочных документов, включая любые поправки и изменения к ним:

ИСО/МЭК 8824-1 Информационные технологии. Нотация абстрактного синтаксиса версии 1 (ASN.1). Часть 1. Спецификация базовой нотации (ISO/IEC 8824-1, Information technology — Abstract Syntax Notation One (ASN.1) — Part 1: Specification of basic notation)

ИСО 10303-1 Системы промышленной автоматизации и интеграция. Представление данных о продукции и обмен данными. Часть 1. Обзор и основные принципы (ISO 10303-1, Industrial automation systems and integration — Product data representation and exchange — Part 1: Overview and fundamental principles)

ИСО 15531-1 Системы промышленной автоматизации и интеграция. Управляющая информация промышленным производством. Часть 1. Общий обзор (ISO 15531-1, Industrial automation systems and integration — Industrial manufacturing management data — Part 1: General overview)

ИСО 15531-42 Системы промышленной автоматизации и интеграция. Управляющая информация промышленным производством. Часть 42. Модель времени (ISO 15531-42:2005, Industrial automation systems and integration — Industrial manufacturing management data — Part 42: Time model)

3 Термины, определения и сокращения

3.1 Термины и определения

В настоящем стандарте использованы следующие термины с соответствующими определениями.

3.1.1 **аксиома** (axiom): Точно сформулированное аналитическое выражение на формальном языке, устанавливающее ограничения к интерпретации символов в словаре языка.

[ИСО 18629-1:2004, статья 3.1.1]

Издание официальное

3.1.2 **консервативное определение** (conservative definition): Определение, устанавливающее необходимые и достаточные условия для полного соответствия термина, а также не позволяющее выводить новые умозаключения из теории.

[ИСО18629-1:2004, статья 3.1.2]

3.1.3 **основная теория** (core theory): Набор предикат, функциональных символов и постоянных величин, ассоциированных с некоторыми аксиомами и элементарными понятиями онтологии.

3.1.4 **данные** (data): Представление информации в формальном виде, подходящем для ее передачи, интерпретации или обработки людьми или на электронно-вычислительных машинах.

[ИСО 10303-1]

3.1.5 **установленная лексика** (defined lexicon): Набор символов в нелогической лексике, обозначающих установленные понятия.

Примечание — Установленные лексикой символы подразделяют на постоянные, функциональные и относительные.

Пример — Термины с консервативными определениями.

[ИСО 18629-1:2004, статья 3.1.5]

3.1.6 **расширение** (extension): Расширение ядра PSL, содержащее дополнительные аксиомы.

Примечание 1 — Ядро PSL представляет собой относительно простой набор аксиом, достаточный для представления широкого круга основных процессов. Однако для представления более сложных процессов требуются дополнительные ресурсы, отсутствующие в ядре PSL. Ядро PSL с каждым понятием следует использовать для описаний того или иного процесса, а для описания разнообразных модульных расширений следует использовать расширение и дополнения ядра PSL. В этом случае пользователь может использовать такой язык, который соответствует требованиям к выразительности.

Примечание 2 — Все расширения являются теориями ядра или дефинициальными расширениями.

[ИСО 18629-1:2004, статья 3.1.9]

3.1.7 **грамматика** (grammar): Правила совместного использования логических символов и словарных терминов для составления точно сформулированных аналитических выражений.

[ИСО 18629-1:2004, статья 3.1.10]

3.1.8 **индивид** (individual): Отдельный логический элемент домена, являющийся неделимым без потери идентичности.

Пример — Отдельная постоянная величина (константа) — это символ, используемый для ссылки на какой-либо отдельный объект, представляющий логический эквивалент «имени» в обиходном языке. В логике первого порядка аргументы предикатов (утверждений) всегда являются отдельными константами.

Примечание 1 — Дополнительная информация приведена в [5].

Примечание 2 — Данный термин обычно используется в формальной логике.

Примечание 3 — В логике первого порядка только индивидуумы являются отдельными константами.

3.1.9 **информация** (information): Факты, концепции или инструкции.

[ИСО 10303-1]

3.1.10 **интерпретация** (interpretation): Предметная область и присвоение истинных значений (TRUE или FALSE) всем предложениям в теории.

Примечание — Пример интерпретации приведен в приложении С.

3.1.11 **язык** (language): Сочетание лексики и грамматики.

[ИСО 18629-1:2004, статья 3.1.12]

3.1.12 **лексика** (lexicon): Набор символов и терминов.

Примечание — Лексика состоит из логических (например, булевы выражения и квантификаторы) и нелогических символов. В комплексе стандартов ИСО 18629 нелогическая часть лексики состоит из выражений (констант, функциональных символов и реляционных символов), необходимых для представления основных понятий онтологии.

[ИСО 18629-1:2004, статья 3.1.13]

3.1.13 **линейное упорядочение** (linier ordering): Совокупность элементов с бинарным отношением между любыми двумя элементами, являющаяся переходной, нерефлексивной и ассиметричной.

Пример — Отношение «меньше чем» в математике: 3<5.

3.1.14 **производство** (manufacturing): Функция или действие, предусматривающие перевод или превращение материала из сырья или заготовки в завершенное состояние.

[ИСО 15531-1:2004, статья 3.6.22]

3.1.15 **производственный процесс** (manufacturing process): Структурированный комплекс видов деятельности или работ, выполняемых с материалом для его перевода из сырья или заготовки в завершенное состояние.

Примечание — Производственные процессы могут быть представлены в виде технологической схемы процесса, схемы движения продукта, в виде табличной схемы или схемы фиксированного расположения. К планируемым производственным процессам могут относиться изготовление продукта для складирования, на заказ и для сборки на заказ и т. д., основанным на стратегическом использовании и размещении материально-производственных запасов.

[ИСО 15531-1:2004, статья 3.6.25]

3.1.16 **модель** (model): Сочетание набора элементов и истинного назначения, удовлетворяющее всем правильно построенным формулировкам в теории.

Примечание 1 — В настоящем стандарте определение термина «модель» отличается от используемого в научной и другой литературе: если предложение является верным в определенной интерпретации, то можно сказать, что интерпретация — это модель предложения. Виды семантик, представленных в настоящем стандарте, часто называют теоретически смоделированными семантиками.

Примечание 2 — Модель обычно представляют в виде совокупности дополнительных структур (частично упорядоченных, в качестве структурного или векторного пространства). В этом случае модель определяет значения для терминологии и понятия истины для предложений языка в условиях данной модели. Задавая модель, основной набор аксиом математических структур, используемый в наборе аксиом, используют как основу для определения понятий, представленных в терминах языка, и их логических взаимосвязей, в результате чего набор моделей создает формальные семантики онтологии.

[ИСО 18629-1:2004, статья 3.1.16]

3.1.17 **онтология** (ontology): Лексика специализированной терминологии, дополненная необходимой спецификацией значений терминов.

Примечание 1 — Онтология — это структурированный набор относительных терминов, представленный с описанием значений терминов на формальном языке. Описание значения объясняет, как и почему термины соотносятся, и определяет условия сегментирования и структурирования набора терминов.

Примечание 2 — Основополагающим компонентом языка технологических спецификаций ИСО 18629 является онтология. Примитивные концепции в онтологии, соответствующей определению ИСО 18629, достаточны для описания основных производственных и инженерных процессов, а также бизнес-процессов.

Примечание 3 — Основное внимание онтологии направлено не только на термины, но и на их значения. Произвольный набор терминов включен в онтологию, но эти термины могут приниматься только в том случае, если их значения согласованы. Такие предполагаемые семантики терминов могут быть утверждены и использованы.

Примечание 4 — Любой термин, используемый без точного определения, может быть причиной неясности и путаницы. Сложность онтологии заключается в том, что структура нуждается в создании терминов, имеющих точное значение. Для онтологии, соответствующей определению ИСО 18629, необходимо предоставить математически строгую характеристику информационного процесса, а также четкое выражение основных логических свойств этой информации на языке, указанном в ИСО 18629.

[ИСО 18629-1:2004, статья 3.1.17]

3.1.18 **момент времени** (point in time): Местоположение чего-либо достойного внимания в пределах временной области.

Пример 1 — Среда, 15 марта, 2003.
Пример 2 — 9.30 утра.

[ИСО 15531-42:2005, статья 3.1.13]

3.1.19 **примитивная концепция** (primitive concept): Лексический термин, не имеющий консервативного определения.

[ИСО 18629-1:2004, статья 3.1.19]

3.1.20 **примитивная лексика** (primitive lexicon): Набор символов в нелогическом словаре, обозначающих элементарные понятия.

Примечание — Примитивная лексика включает в себя постоянные, функциональные и реляционные символы.

[ИСО 18629-1:2004, статья 3.1.20]

3.1.21 **процесс** (process): Структурированный ряд видов деятельности, включающий в себя различные сущности предприятия, предназначенный и организованный для достижения конкретной цели.

Примечание — Данное определение аналогично определению, приведенному в ИСО 10303-49. Тем не менее ИСО 15531 нуждается в понятии структурированного набора деятельностей без какого-либо предопределенного отношения ко времени или этапам. С точки зрения управления потоком некоторые свободные процессы могут требовать синхронизации в отношении цели, хотя в действительности они ничего не выполняют (задачи-призраки).

[ИСО 15531-1:2004, статья 3.6.29]

3.1.22 **продукт** (product): Изделие, материал или вещество, изготовленное в процессе производства.

[ИСО 10303-1]

3.1.23 **теория доказательств** (proof theory): Совокупность теорий и лексических элементов, необходимых для интерпретации семантики языка.

Примечание — Теория доказательств состоит из трех компонентов: ядра PSL, внешнего ядра и расширений.

[ИСО 18629-1:2004, статья 3.1.25]

3.1.24 **ядро PSL** (PSL Core): Набор аксиом для понятий деятельности, события деятельности, момента времени и объекта.

Примечание — Мотивацией для ядра PSL является наличие любых двух приложений, имеющих отношение к процессу, которые должны совместно использовать упомянутые аксиомы с целью обмена информацией о процессе. Поэтому ядро PSL является адекватным для описания основных концепций производственных процессов. Следовательно, эта характеристика основных процессов имеет несколько допущений в отношении их характеристик, за исключением тех, которые необходимы для описания процессов. Поэтому ядро PSL ограничено с точки зрения выражения логической возможности. При этом ядро PSL обеспечивает определение многих вспомогательных понятий, которые необходимы для описания всех интуитивных понятий в производственном процессе.

[ИСО 18629-1:2004, статья 3.1.26]

3.1.25 **теория** (theory): Набор аксиом и определений, относящийся к данному понятию или набору понятий.

Примечание — Данное определение отражает подход искусственного интеллекта, где теория — это набор предположений, на которых основано значение соответствующего понятия.

[ИСО 18629-1:2004, статья 3.1.31]

3.1.26 **область обсуждения** (universe of discourse): Совокупность конкретных или абстрактных вещей, относящихся к области реального мира, выбранных в соответствии с интересом, который они представляют для системы, подлежащей моделированию, и ее окружения.

[ИСО 15531-1:2004, статья 3.6.50]

3.2 Сокращения

В настоящем стандарте применены следующие сокращения:

BNF — форма Бэкуса-Наура (Backus-Naur form);

KIF — формат обмена знаниями (Knowledge Interchange Format);

PSL — язык спецификаций процесса (Process Specification Language).

4 Общие положения комплекса стандартов ИСО 18629

Комплекс стандартов ИСО 18629 устанавливает требования к лексике, онтологии и грамматике языка, используемого для описания производственных процессов.

Примечание 1 — PSL — это язык, используемый для спецификации производственных процессов, основанный на математически точно определенных словаре и грамматике. Он отличается от языка, используемого для описаний схем и продукта в ИСО 10303, ИСО 13584 и ИСО 15926, а также от языка характеризации процессов, приведенного в ИСО 15531, но тесно с ним связан и дополняет его. При обмене информацией между двумя процессами язык PSL определяет каждый процесс, независимо от его поведения (режима работы). Например, объект, рассматриваемый как ресурс для одного процесса, считают таким же объектом, даже если его рассматривают в качестве продукта второго процесса.

Примечание 2 — PSL основан на логике первого порядка и использует другие методы спецификации семантики, используемые в ИСО 10303. Значения понятий, определяемых в рамках PSL, вытекают из набора аксиом и определений, представленных каждым расширением ядра PSL. Примечания и примеры, приведенные в стандартах, должны способствовать пониманию языка.

В частях с 11 по 19 комплекса стандартов ИСО 18629 приведены основные теории, необходимые для представления точных определений и аксиом элементарных концепций. Основные теории обеспечивают точные семантические переводы между разными схемами.

Части с 11 по 19 распространяются на:

- представление основных элементов языка;
- обеспечение стандартизованных наборов аксиом, соответствующих интуитивным элементарным семантическим концепциям описания основных процессов;
- свод правил, необходимых для разработки других основных теорий или расширений в соответствии с ядром языка спецификации процесса.

Данные части комплекса стандартов ИСО 18629 не распространяются на представление информации, относящейся к концепциям, не являющимся частью основных теорий.

5 Синтаксис спецификации языка

В настоящем разделе приведено определение и описание языка, установленного в комплексе стандартов ИСО 18629, с помощью расширенной формы Бэкуса-Наура (BNF) (см. приложение D). Основой синтаксиса, описание которого приведено ниже, является синтаксис формата обмена знаниями (KIF), представленного в приложении B.

Примечание — Дополнительная информация приведена в [7] и [8].

5.1 Основные символы и синтаксические категории

Основные символы, которые следует использовать для спецификации описания любого процесса на языке PSL, определяют с помощью следующей формы Бэкуса-Наура:

- <uc-letter> ::=A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W |X | Y | Z
- <lc-letter> ::= a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z
- <letter> ::= <uc-letter> | <lc-letter>
- <digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
- <oper> ::= - | ~ | # |$ | * | + | /
- <punct> ::= _ | - | ~ | ! | @ | # |$ | % | ^ | & | * | (|) | + | = | ` | : | ; | ' | < | > | , | . | ? | / | | | [|] | { | } | <space>

Выражение представляет собой любую последовательность основных символов.

Существуют пять следующих основных категорий выражения:

- <b-con> ::= {<lc-letter> | <digit>} {<letter> | <digit>}* {{ _ | - } {<letter> | <digit>}}*
- <b-var> ::= ?<b-con>[']
- <b-func> ::= {<oper> | <uc-letter>} {<letter> | <digit>}* {{ _ | - } {<letter> | <digit>}}*
- <b-pred> ::= {<uc-letter>} {<letter> | <digit>}* {{ _ | - } {<letter> | <digit>}}*
- <doc-string> ::= " {<letter> | <digit> | <punct> | \" | \\ }* "

Выражение, выведенное из нетерминального символа <b-con>, является последовательностью буквенно-цифровых символов, тире и символов подчеркивания. Последовательность начинается с буквы или цифры нижнего регистра, в которой каждое тире и символ подчеркивания ограничены буквой или цифрой. Выражение <b-var> — это результат задания префикса <b-con> с вопросительным знаком и необязательно с присоединением на конце одинарной кавычки (штриха). Выражение <b-func> аналогично выражению <b-con> за исключением того, что оно должно начинаться либо с <oper>, <punc>, либо с буквы верхнего регистра, а выражение <b-pred> аналогично выражению <b-con> за исключением того, что оно должно начинаться с буквы верхнего регистра (таким образом, каждое выражение <b-pred> является выражением <b-func>). Выражение <doc-string> — это результат заключения в кавычки любой последовательности символов; двойные кавычки и обратная (правая) косая черта могут использоваться до тех пор, пока им предшествует обратная (правая) косая черта.

5.2 Лексика

Язык первого порядка L языка спецификации процесса состоит из терминов словаря и грамматики. Лексика обеспечивает основные слова языка, а грамматика устанавливает, как могут быть использованы лексические элементы для построения сложных, правильно сформированных выражений языка.

Словарь PSL (lexicon λ) состоит из:

- выражения <space>, (,), not, and, or, implies, iff, forall и exists;
- нетерминальной рекурсивной совокупности V_λ выражений <b-var>, известных как переменные λ;

- рекурсивной совокупности C_λ выражений <b-con>, известных как переменные λ, которые включают в себя по меньшей мере последовательности inf-, inf+, max- и max+;

- рекурсивной совокупности F_λ выражений <b-func>, известных как функциональные символы λ, которые включают в себя по меньшей мере последовательности beginof и endof;

- рекурсивной совокупности P_λ выражений <b-pred>, известных как предикаты λ, которые включают в себя по меньшей мере последовательности activity (действие), activity-occurrence (появление действия), object (объект), timepoint (момент времени), is-occuring-at (происходит в момент времени), occurrence-of (событие чего-либо), exists-at (существует в момент времени) и participates-in (участвует в чем-либо).

5.3 Грамматика

В словаре lexicon λ грамматику на основе λ устанавливают в следующей форме Бэкуса-Наура:

- <con> ::= элемент C_λ;
- <var> ::= элемент V_λ;
- <func> ::= элемент F_λ;
- <pred> ::= элемент P_λ;
- <term> ::= <atomterm> | <compterm>;
- <atomterm> ::= <var> | <con>;
- <compterm> ::= (<func> <term>);
- предложение <sentence> ::= <atomsent> | <boolsent> | <quantsent>;
- <atomsent> ::= (<pred> <term>+) | (<term> <term>);
- <boolsent> ::=(not <sentence>) | (and <sentence> <sentence>+) | (or <sentence> <sentence>+) | (implies <sentence> <sentence>) | (iff <sentence> <sentence>);
- <quantsent> ::= ({forall | exists} {<var> | (<var>+) } <sentence>);
- <psl-sentence> ::= <sentence>.

5.4 Язык

Язык PSL L_λ, созданный на основе словаря λ, представляет собой совокупность выражений, которые могут быть выведены из нетерминального предложения PSL <psl-sentence> с использованием вышеуказанной грамматики. Элементы L_λ являются предложениями (sentences) L_λ. Ограничения, установленные пользователем, должны быть созданы с использованием языка PSL.

Примечание — Примеры ограничений, установленных пользователем, приведены в приложении С.

6 Основные элементы ядра PSL

В настоящем стандарте определены элементы, которые составляют ядро PSL.

6.1 Основные свойства

Основными свойствами языка являются:

- элементарная лексика;
- определенная лексика с поддерживающими определениями;
- аксиомы.

Так как данные понятия являются элементами ядра языка, нет необходимости применять дополнительную теорию.

Смысловое значение следующих терминов следует либо из аксиом, либо из дополнительных определений.

6.2 Элементарный словарь ядра PSL

В элементарный словарь входят:

- элементарные категории;
- индивиды;
- элементарные отношения;
- элементарные функции.

Примечание — В логических элементарных концепциях не устанавливают формальные определения в пределах онтологии.

6.2.1 Элементарные категории ядра PSL

В настоящем пункте приведены элементарные категории с неформальными описаниями.

6.2.1.1 Деятельность (activity)

Нотация формата обмена знаниями:

(activity ?a)

Неформальное описание:

Деятельность (activity ?a) является истинной (TRUE) в интерпретации ядра PSL только в том случае, если ?a является элементом множества деятельностей в предметной области этой интерпретации.

6.2.1.2 Проявление деятельности наступление события (activity_occurrence)

Нотация формата обмена знаниями: (activity_occurrence ?occ)

Неформальное описание:

Проявление деятельности (activity_occurrence ?occ) является истинным (TRUE) в интерпретации ядра PSL только в том случае, если ?occ является элементом множества появлений деятельности в предметной области этой интерпретации. Проявление деятельности ассоциируется с однозначной деятельностью, начинающейся и завершающейся в заданные моменты времени.

Примечание — Проявление деятельности не является экземпляром деятельности, хотя может существовать деятельность, которая не имеет проявлений; все проявления деятельности должны быть ассоциированы с конкретной деятельностью.

Пример — Деятельность, обозначенная термином «красить дом, банка краски» (paint House1 Paintcan1), является экземпляром класса деятельности окрашивания (Painting).

Painting (paint House1 Paintcan1)

Существуют многочисленные различные проявления этого экземпляра:

(occurrence_of Occ1 (paint House1 Paintcan1))

(occurrence_of Occ2 (paint House1 Paintcan1))

(=(beginof Occ1) 1100)

(=(endof Occ1) 1200)

(=(beginof Occ2) 1500)

(=(endof Occ2) 1800)

Также существует экземпляр класса деятельности покраски (Painting).

(Painting (paint House1 Paintcan2)), не имеющий появлений.

6.2.1.3 Объект (object)

Нотация формата обмена знаниями:

(object ?x)

Неформальное описание:

Объект (object ?x) является истинным (TRUE) в интерпретации ядра PSL только в том случае, если ?x является элементом множества объектов в предметной области этой интерпретации. Объектом может быть все что угодно, за исключением момента времени и проявления деятельности.

Примечание — Интуитивно предполагается, что объект — это конкретная или абстрактная вещь, которая может участвовать в деятельности. Объекты могут начать существование (быть созданы) и прекратить существование (использоваться как ресурс) в определенные моменты времени. В этом случае объект имеет моменты начала и конца существования. В некоторых контекстах рекомендуется рассматривать некоторые обычные объекты, как не имеющие таких моментов.

Пример 1 — Наиболее типичными примерами объектов являются обычные вещи, например народ, стулья, кузов машины, станки с ЧПУ. Абстрактные объекты, например числа, также являются объектами.

Пример 2 — Некоторые объекты, например числа, не имеют моментов начала и конца существования.

6.2.1.4 Момент времени (timepoint)

Нотация формата обмена знаниями:

(timepoint ?t)

Неформальное описание:

Момент времени (timepoint ?t) является истинным (TRUE) в интерпретации ядра PSL только в том случае, если ?t является элементом множества моментов времени в предметной области этой интерпретации. Моменты времени образуют линейное упорядочение.

6.2.2 Индивиды ядра PSL

6.2.2.1 Отрицательная бесконечность (inf-)

Неформальное описание:

Отрицательная бесконечность (= ?t inf-) является истинной (TRUE) в интерпретации ядра PSL только в том случае, если ?t является однозначным моментом времени, который наступает раньше всех остальных моментов в линейном упорядочении моментов времени в предметной области этой интерпретации.

Примечание — Отрицательная бесконечность (inf-) необходима для определения объектов, которые не были созданы в пределах предметной области этой интерпретации.

6.2.2.2 Положительная бесконечность (inf+)

Неформальное описание для (inf+):

Положительная бесконечность (= ?t inf+) является истинной (TRUE) в интерпретации ядра PSL только в том случае, если ?t является однозначным моментом времени, который наступает после всех остальных моментов в линейном упорядочении моментов времени в предметной области этой интерпретации.

Примечание — Положительная бесконечность (inf+) необходима для определения объектов, которые не были удалены в пределах предметной области этой интерпретации.

6.2.3 Элементарные отношения ядра PSL

6.2.3.1 Перед (before)

Нотация формата обмена знаниями:

(before ?t1 ?t2)

Неформальное описание:

Отношение (before ?t1 ?t2) является истинным (TRUE) в интерпретации ядра PSL только в том случае, если ?t1 наступает раньше, чем ?t2 в линейном упорядочении моментов времени в этой интерпретации.

Примечание — В ядре PSL множество моментов времени не является плотным (между любыми двумя разными моментами времени имеется третий момент времени), несмотря на то что множество моментов времени является бесконечным. Свойство плотности может быть добавлено пользователем в качестве дополнительного постулата (критерия). Интервалы времени не являются примитивами ядра PSL. Интервалы могут быть определены в отношении моментов времени и действий.

Пример — Концепция интервалов, основанная на моментах времени, является основополагающим элементом ИСО 15531-42 (модель времени) и ИСО 10303-41 (описание и поддержка продукта).

6.2.3.2 Проявление (occurrence_of)

Нотация формата обмена знаниями:

(occurrence_of ?occ ?a)

Неформальное описание:

(Occurrence_of ?occ ?a) является истинным (TRUE) в интерпретации ядра PSL только в том случае, если ?occ является частным проявлением действия ?a. (Occurrence_of) — это основное отношение между деятельностью и проявлениями деятельности. Каждое проявление деятельности ассоциируется с однозначным действием. Деятельность может не иметь проявлений или множественных вхождений.

6.2.3.3 Участвует в (participates_in)

Нотация формата обмена знаниями:

(participates_in ?x ?occ ?t)

Неформальное описание:

Отношение (participates_in ?x ?occ ?t) является истинным (TRUE) в интерпретации ядра PSL только в том случае, если ?x играет роль, которая предварительно не задана в событии проявления деятельности ?occ в момент времени ?t в этой интерпретации. Объект может участвовать в проявлении деятельности только в те моменты времени, в которые объект существует, а деятельность происходит.

6.2.4 Элементарные функции ядра PSL

6.2.4.1 Начинать (beginof)

Нотация формата обмена знаниями:

(beginof ?x)

Неформальное описание:

Областью функции (beginof) является совокупность начала наступления деятельности и объектов.

Если ?x — это начало наступления деятельности в предметной области интерпретации ядра PSL, то (beginof ?x) имеет значение x только в том случае, если t является моментом времени, в который начинается деятельность ?x.

Пример — (= 10 (beginof milling_occurrence)) представляет собой действие фрезерования, начинающееся в момент времени 10.

Если ?x — это объект в предметной области интерпретации ядра PSL, то (beginof ?x) имеет значение x только в том случае, если t является моментом времени, в который объект ?x может принять участие в деятельности.

Пример — (= 15 (beginof screw)) представляет собой случай, когда буравчик объекта может принять участие в действии в момент времени 15.

6.2.4.2 Заканчивать (endof)

Нотация формата обмена знаниями:

(endof?x)

Неформальное описание:

Областью функции endof является совокупность начала наступления деятельности и объектов.

Если ?x — это начало наступления деятельности в предметной области интерпретации ядра PSL, то (endof ?x) имеет значение x только в том случае, если t является моментом времени, в который заканчивается деятельность ?x.

Пример — (= 25 (endof milling_occurrence)) представляет собой случай, когда действие фрезерования заканчивается в момент времени 25.

Если ?x является объектом в предметной области интерпретации ядра PSL, то (endof ?x) имеет значение x только в том случае, если t является моментом времени, в который объект ?x больше не может принимать участие в действии.

Пример — (= 30 (endof screw)) представляет собой случай, когда буравчик объекта не может принимать участие в действии в момент времени 30.

6.2.5 Теории, требуемые ядром PSL

Для элементарной лексики не требуется применения других теорий.

6.3 Определенная лексика ядра PSL

6.3.1 Формальные определения ядра PSL

Примечание — Термины 'defrelation', 'exists', 'forall', 'and', 'or', 'not', '=', '⇔' и 'implies' определены в Руководстве формата обмена знаниями [8].

6.3.1.1 Между (between)

Определение 1: момент времени (Timepoint t2) наступает между моментами времени t1 и t3 только в том случае, если t1 наступает перед t2, а t2 — перед t3.

```
(forall (?t1 ?t2 ?t3) (iff (between ?t1 ?t2 ?t3)
        (and (before ?t1 ?t2) (before ?t2 ?t3)))))
```

6.3.1.2 Перед или равен (beforeEq)

Определение 2: момент времени (Timepoint t1) наступает перед или одновременно с t2 только в том случае, если t1 наступает перед или одновременно с t2.

```
(forall (?t1 ?t2) (iff (beforeEq ?t1 ?t2)
        (and (timepoint ?t1) (timepoint ?t2)
                (or (before ?t1 ?t2) (= ?t1 ?t2))))))
```

6.3.1.3 Между или равен (betweenEq)

Определение 3: момент времени (Timepoint t2) наступает между или одновременно с моментами времени t1 и t3 только в том случае, если t1 наступает перед или одновременно с t2, а t2 — перед или равен t3.

```
(forall (?t1 ?t2 ?t3) (iff (betweenEq ?t1 ?t2 ?t3)
        (and (beforeEq ?t1 ?t2)
                (beforeEq ?t2 ?t3)))))
```

6.3.1.4 Существует в момент времени (exists_at)

Определение 4: объект существует в момент времени t1 только в том случае, если t1 расположен между или одновременно с моментами его начала и окончания.

```
(forall (?x ?t) (iff (exists_at ?x ?t)
        (and (object ?x)
                (betweenEq (beginof ?x) ?t (endof ?x))))))
```

6.3.1.5 Происходит в момент времени (is_occurring_at)

Определение 5: действие происходит в момент времени t1 только в том случае, если t1 наступает в моменты начала и окончания этого действия.

```
(forall (?occ ?t) (iff (is-occuring-at ?occ ?t)
        (and (activity_occurrence ?occ)
             (betweenEq (beginof ?occ) ?t (endof ?occ)))))
```

6.4 Аксиомы

Основными аксиомами ядра PSL являются axiom 1 — axiom 17.

Примечание — Термины 'defrelation', 'exists', 'forall', 'and', 'or', 'not', '=', 'iff', и 'implies' определены в Руководстве формата обмена знаниями [8].

6.4.1 Аксиома 1

Отношение before происходит только между моментами времени (timepoints).

```
(forall (?t1 ?t2)
        (implies (before ?t1 ?t2)
                 (and (timepoint ?t1)
                      (timepoint ?t2))))
```

6.4.2 Аксиома 2

Отношение before представляет собой нестрогий порядок.

```
(forall (?t1 ?t2)
        (implies (and (timepoint ?t1)
                      (timepoint ?t2))
                 (or (= ?t1 ?t2)
                     (before ?t1 ?t2)
                     (before ?t2 ?t1)))))
```

6.4.3 Аксиома 3

Отношение before является иррефлексивным.

```
(forall (?t) (not (before ?t ?t)))
```

6.4.4 Аксиома 4

Отношение before является транзитивным (переходным).

```
(forall (?t1 ?t2 ?t3)
        (implies (and (before ?t1 ?t2)
                      (before ?t2 ?t3))
                 (before ?t1 ?t3)))
```

6.4.5 Аксиома 5

Отрицательная бесконечность момента времени (timepoint inf-) существует перед всеми другими моментами времени.

```
(forall (?t)
(implies (and (timepoint ?t) (not (= ?t inf-)))
        (before inf- ?t))
```

6.4.6 Аксиома 6

Перед положительной бесконечностью (inf+) существуют все остальные моменты времени.

```
(forall (?t)
(implies (and (timepoint ?t) (not (= ?t inf+)))
        (before ?t inf+))
```

6.4.7 Аксиома 7

Момент времени между inf- и t наступает при любом моменте времени t, не являющимся отрицательной бесконечностью (inf-).

```
(forall (?t)
        (implies (and (timepoint ?t)
                      (not (= ?t inf-)))
        (exists (?u)
                      (between inf- ?u ?t))))
```

6.4.8 Аксиома 8

Момент времени между inf+ и t наступает при любом моменте времени t, не являющимся положительной бесконечностью (inf+).

```
(forall (?t)
        (implies (and (timepoint ?t)
                      (not (= ?t inf+)))
        (exists (?u)
                      (between ?t ?u inf+))))
```

6.4.9 Аксиома 9

Все описываемые сущности есть либо действия, либо объекты, либо наступление действия, либо момент времени.

```
(forall (?x)
        (or (activity ?x)
            (activity_occurrence ?x)
            (object ?x)
            (timepoint ?x)))
```

6.4.10 Аксиома 10

Объекты, действия, наступления действий и моменты времени являются разными видами сущностей.

```
(forall (?x)
(and (implies  (activity ?x)
                (not (or (activity_occurrence ?x) (object ?x) (timepoint ?x))))
        (implies (activity_occurrence ?x)
                (not (or (object ?x) (timepoint ?x))))
        (implies (object ?x)
                (not (timepoint ?x)))))
```

6.4.11 Аксиома 11

Отношение наступления действия существует только между действиями и наступлениями действий.

```
(forall (?a ?occ)
        (implies (occurrence_of ?occ ?a)
                (and (activity ?a)
                        (activity_occurrence ?occ))))
```

6.4.12 Аксиома 12

Наступление действия является однозначным действием.

```
(forall (?occ ?a1 ?a2)
        (implies (and (occurrence_of ?occ ?a1)
                        (occurrence_of ?occ ?a2))
                (= ?a1 ?a2))))
```

6.4.13 Аксиома 13

Начало и окончание наступления действия или объекта являются моментами времени.

```
(forall (?x)
        (implies (or  (activity_occurrence?x)  (object ?x))
                (and  (timepoint (beginof ?x))
                        (timepoint (endof ?x)))))
```

6.4.14 Аксиома 14

Момент начала каждого действия или объекта наступает перед моментом его окончания или равен моменту его окончания.

```
(forall (?x)
        (implies (or (activity_occurrence ?x )
                (object ?x))
                (beforeEq (beginof ?x) (endof ?x))))
```

Примечание — Если моменты начала и окончания действия являются равными, то поддерживаются приложения программного обеспечения, которые представляют мгновенные события.

6.4.15 Аксиома 15

Отношение участвует в чем-либо (participate_in) может существовать только между объектами, наступлениями действий и моментами времени соответственно.

```
(forall (?x ?occ ?t)
        (implies (participates_in ?x ?occ ?t)
        (and (object ?x)
                (activity_occurrence ?occ)
                (timepoint ?t))))
```

6.4.16 Аксиома 16

Объект может участвовать в наступлении действия только в те моменты времени, в которые он существует, а действие происходит.

```
(forall (?x ?occ ?t)
        (implies (participates_in ?x ?occ ?t)
                 (and (exists_at ?x ?t)
                      (is_occurring_at ?occ ?t))))
```

6.4.17 Аксиома 17

Каждое наступление действия является вхождением действия.

```
(forall (?occ)
        (implies (activity_occurrence ?occ)
                 (exist (?a)
                        (and (activity ?a)
                             (occurrence_of ?occ ?a)))))
```

7 Соответствие ядру PSL

7.1 Соответствие онтологий

Онтология соответствует ядру PSL только в том случае, если концепции, предоставленные онтологией, определены таким образом, что онтология согласуется с аксиомами и определениями ядра PSL, приведенными в разделе 6.

7.2 Соответствие описаний процесса

Описание процесса соответствует ядру PSL только в том случае, если онтология описания процесса соответствует ядру PSL, а грамматика описания процесса соответствует положениям раздела 5.

Приложение А
(обязательное)

Идентификатор ASN.1, присвоенный ИСО 18629-11

Для обеспечения точной идентификации информационного объекта в открытой системе, настоящему стандарту присвоен следующий идентификатор объекта:

ISO standard 18629 part 11 version 1

Смысл этого значения определен в ИСО/МЭК 8824-1 и описан в ИСО18628-1.

Приложение В
(справочное)

Синтаксис формата обмена знаниями и семантика

Подобно машинно-ориентированным языкам, формат обмена знаниями (KIF) имеет два многообразия. В линейном формате обмена знаниями все выражения являются последовательностями символов в коде ASCII, поэтому их можно хранить на устройствах с последовательной передачей данных (например, на магнитных дисках), а также в последовательных средах (например, по телефонным линиям). В структурированном формате обмена знаниями выражения языка являются структурированными объектами. Структурированный формат обмена знаниями имеет специальное применение в коммуникации между программами, работающими в том же самом адресном пространстве [8].

Между двумя многообразиями KIF существует простая корреспонденция. Для каждой строки символов имеется только одна соответствующая списковая структура, а для каждой списковой структуры — только одна соответствующая строка символов.

Сначала определяется соответствие между линейными и структурированными формами языка, после чего данные обрабатываются только в структурированной форме.

В.1 Линейный формат обмена знаниями

Алфавит линейного формата обмена знаниями состоит из 128 символов набора кода ASCII. Некоторые из этих символов имеют стандартные печатные представления. Символами, имеющими стандартные печатные представления (98 из 128), являются следующие:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9 ( ) [ ] - - Ø ! ?
= + - * / `` & o ~ ' ` " ù @ # $ % : ; , . ! ?

Формат обмена знаниями основан на языке приложения Lisp (LISP) и наследует от него свой синтаксис. Отношение между линейным и структурированным KIF упрощается с помощью считывающего устройства Common Lisp. В частности, строка символов ASCII образует формальное выражение в линейном KIF только в том случае, если:

1) она является приемлемой для считывающего устройства Common Lisp;

2) структура, созданная считывающим устройством Common Lisp, является формальным выражением структурированного KIF (согласно положениям следующего раздела).

В.2 Структурированный KIF

В структурированном KIF нотация word является примитивным элементом. Выражение является либо словом, либо конечной последовательностью выражений. В данном случае используют закрывающие скобки, чтобы ограничить элемент в составном выражении:

<word> ::= a primitive syntactic object (элементарный синтаксический объект)
<expression> ::= <word> | (<expression>*)

Все слова подразделены на категории, приведенные ниже. Эта категоризация является непересекающейся и исчерпывающей. Каждое слово входит в состав только одной категории. Категории, определенные в настоящем разделе, используются также в грамматических правилах следующих таблиц.

<indvar> ::= a word beginning with the character ?
<seqvar> ::= a word beginning with the character @
<termop> ::= listof | setof | quote | if | cond | the | setofall | kappa | lambda
<sentop> ::= = | /= | not | and | or | implies | <= | iff | forall | exists
<ruleop> ::= =>> | <<= | consis
<defop> ::= defobject | defunction | defrelation | := | :implies | :&
<objconst> ::= a word denoting an object
<funconst> ::= a word denoting a function
<relconst> ::= a word denoting a relation
<logconst> ::= a word denoting a truth value

Из этих основных категорий строят более сложные категории — переменные (variables), операторы (operators) и константы (constants).

<variable> ::= <indvar> | <seqvar>
<operator> ::= <termop> | <sentop> | <ruleop> | <defop>
<constant> ::= <objconst> | <funconst> | <relconst> | <logconst>

variable — это слово, в котором первым символом является ? или @. Переменную, начинающуюся с ?, называют отдельной переменной. Переменную, начинающуюся с @, называют переменной последовательнос-

ти. Отдельные переменные используются в квантификации по отдельным объектам. Переменные последовательностей используются в квантификации по последовательностям объектов.

Операторы используются для образования сложных выражений разных типов. В KIF имеется четыре типа операторов — term operators (термовые операторы терминов), sentence operators (операторы предложений), rule operators (операторы правил) и definition operators (операторы определений). Операторы терминов используют для формирования сложных термов, операторы предложений — для формирования сложных предложений, операторы правил — для формирования правил, операторы определений — для формирования определений.

Константой является любое слово, не являющееся переменной величиной или оператором. В KIF имеются четыре категории констант: object constants (константы объектов), function constants (константы функции), relation constants (константы отношений) и logical constants (логические константы).

Константы объектов обозначают отдельные объекты, константы функций — функции для упомянутых объектов, константы отношений — отношения; логические константы выражают условия в отношении слов и являются либо истинными, либо ложными.

Некоторые константы являются базовыми — их тип и смысловое значение фиксируются в определении KIF. Все другие постоянные величины не являются базовыми — пользователь языка должен выбрать тип и смысловое значение. Все числа, символы и строки в KIF являются базовыми константами, характеристики остальных базовых констант должны соответствовать приведенным в настоящем стандарте.

Формат обмена знаниями является необычным логическим языком. Невозможно установить по орфографии категорию небазовой константы (т. е. определить, является ли она объектом, функцией, отношением или логической постоянной величиной). Пользователь должен сам выбрать категорию каждой небазовой константы. Пользователь не должен пояснять константы. Однако категория константы устанавливает, как ее можно использовать для формирования выражений, и эта категория может быть установлена в зависимости от этого использования. Так как константу используют особым образом, ее категория становится фиксированной.

В языке имеются четыре специальных типа выражений: terms (термины), sentences (предложения), rules (правила) и definitions (определения). Термины используют для обозначения объектов в области, подлежащей описанию, предложения — для выражения фактов этой области, правила — для выражения шагов логического вывода, а определения — для определения постоянных величин. Формами являются предложения, правила и определения.

Имеются следующие десять типов терминов в KIF: individual variables, object constants, function constants, relation constants, functional terms, list terms, set terms, quotations, logical terms и quantified terms (индивидуальные переменные, объектные константы, функциональные константы, константы отношений, функциональные термины, списочные термины, термины множества, цитаты, логические термины и количественные термины).

Ранее в настоящем стандарте были рассмотрены следующие отдельные константы, константы объектов, константы функций и константы отношений:

<term> ::= <indvar> | <objconst> | <funconst> | <relconst> | <funterm> | <listterm> | <setterm> |
<quoterm> | <logterm> | <quanterm>
<listterm> ::= (listof <term>* [<seqvar>])
<setterm> ::= (setof <term>* [<seqvar>])
<funterm> ::= (<funconst> <term>* [<seqvar>])
<quoterm> ::= (quote <expression>)
<logterm> ::= (if <sentence> <term> [<term>]) | (cond (<sentence> <term>) ... (<sentence> <term>))
<quanterm> ::= (the <term> <sentence>) | (setofall <term> <sentence>) |
(kappa (<indvar>* [<seqvar>]) <sentence>*) | (lambda (<indvar>* [<seqvar>]) <term>)

Функциональный термин (functional term) состоит из константы функции и произвольного числа терминов аргументов, ограниченных переменной необязательной последовательности. Синтаксического ограничения к числу терминов аргументов нет — константа одной и той же функции может быть применена к разному числу аргументов; ограничения арности в KIF обрабатываются семантически.

Списочный термин (list term) состоит из оператора списка (listof) и конечного списка терминов, завершенных необязательной переменной последовательностью.

Термин множество (set term) состоит из оператора множества (setof) и конечного списка терминов, завершенных необязательной переменной последовательностью.

Цитаты (заключенные в кавычки; quotations) включают в себя оператор кавычек и выражение произвольного списка. Вложенное выражение может иметь структуру произвольного списка; оно не нуждается в том, чтобы быть формальным выражением в KIF.

Считывающее устройство LISP преобразует строки формы ‘σ в (σ в кавычках).

Логические термины (logical terms) включают в себя операторы if и cond. Форма if предусматривает тестирование единичного условия, а форма cond предусматривает тестирование последовательности условий.

Количественные термины (термины под знаком квантора; quantified terms) включают в себя операторы the, setofall, kappa и lambda. Указатель (designator) состоит из оператора, термина и предложения. Термин, образующий множество (set-forming term), состоит из оператора setoff, термина и предложения. Термин, образующий отношение (relation-forming term), состоит из буквы каппа, списка переменных и предложения. Термин, образую-

щий функцию (function-forming term), состоит из буквы лямбда, списка переменных и термина. Нет необходимости в использовании каппы и лямбды — обе эти буквы греческого алфавита могут быть определены в терминах setof; их включают в KIF для удобства.

Следующие формы Бэкуса-Наура определяют множество формальных предложений в KIF. Имеется шесть типов предложений.

<sentence> ::= <logconst> | <equation> | <inequality> | <relsent> | <logsent> | <quantsent>
<equation> ::= (= <term> <term>)
<inequality> ::= (/= <term> <term>)
<relsent> ::= (<relconst> <term>* [<seqvar>]) | (<funconst> <term>* <term>)
<logsent> ::= (not <sentence>) |
(and <sentence>*) |
(or <sentence>*) |
(implies <sentence>* <sentence>) |
(<= <sentence> <sentence>*) |
(iff <sentence> <sentence>)
<quantsent> ::= (forall <indvar> <sentence>) |
(forall (<indvar>*) <sentence>) |
(exists <indvar> <sentence>) |
(exists (<indvar>*) <sentence>)

Равенство (equation) состоит из оператора «=» и двух терминов.

Неравенство (inequality) состоит из оператора «/=» и двух терминов.

Связанное предложение состоит из константы отношений и произвольного числа терминов аргументов, завершенных необязательной переменной последовательности. Так же как с функциональными терминами, в предложении отношений нет синтаксического ограничения к числу терминов аргументов — одна и та же константа отношений может быть применена к любому числу аргументов.

Синтаксис логических предложений logical sentences зависит от используемого логического оператора. Предложение, использующее оператор not, называют отрицанием (negation). Предложение, использующее оператор and, называют конъюнкцией (conjunction), а аргументы — конъюнктами (conjuncts). Предложение, использующее оператор or, называют дизъюнкцией (disjunction), а аргументы — дизъюнктами (disjuncts). Предложение, использующее оператор implies, называют импликацией (implication); все его аргументы, за исключением последнего, называют antecedents; а последний аргумент — consequent. Предложение, использующее оператор «<=», называют обратной импликацией; ее первый аргумент называют consequent, а остальные аргументы — (antecedents). Предложение, использующее оператор iff, называют эквивалентностью (equivalence).

Имеются два типа предложений под знаком квантора (quantified sentences) — универсально квантифицируемое предложение (universally quantified) задают с помощью сигнала путем использования оператора forall и экзистенционально квантифицируемое предложение (existentially quantified sentence) задают с помощью сигнала путем использования оператора exists.

B.3 Семантика

Интерпретацией языка KIF первого порядка является кортеж $I = \langle O, ext, \sigma, \tau \rangle$, задаваемый следующим образом.

B.3.1 Функция расширения

В соответствии с интерпретацией «I» функция расширения ext имеет следующие свойства:

- = $E \cup R$;
- = $E \cap R = \varnothing$;
- = If $s \in E$, then ext (s) = $\varnothing$;
- = If $s \in R$, then ext (s) $\subseteq O^*$.

B.3.2 Функция семантического значения для терминов

В соответствии с интерпретацией «I» функция семантического значения σ должна иметь свойства, указанные в следующих пунктах.

B.3.2.1 Семантическое значение переменных объекта (Object Variables)

Семантическим значением переменой объекта ν является объект, назначаемый этой переменой в интерпретации «I»:

$\sigma(\nu) \in O$

B.3.2.2 Семантическое значение постоянных величин (Constants)

Семантическим значением константы s является объект, назначаемый этой постоянной величине в интерпретации «I»:

$\sigma(s) \in O$.

B.3.2.3 Семантическое значение функциональных терминов (Function Terms)

Если σ(F) является функциональным значением, то σ((Ft$_1$ K t$_n$)), является однозначным e ∈ E, так что в интерпретации I cc (σ(t$_1$), K σ(t$_n$), e) ∈ ext (σ(F))

следовательно

σ((Ft$_1$ K t$_n$)) = σ(F).

B.3.3 Функция выполнения для предложений (Sentences)

В соответствии с интерпретацией «I» функция выполнения τ имеет свойства, заданные следующим образом.

B.3.3.1 Равенства

Равенство является истинным только в том случае, если термины в равенстве ссылаются на один и тот же объект в предметной области интерпретации «I»:

$$\tau((= t_1\ t_2)) = \begin{cases} \text{true } \sigma(t_1) = \sigma(t_2) \text{ или} \\ \text{false} \end{cases}$$

B.3.3.2 Атомарное высказывание

Атомарное высказывание является истинным только в том случае, если кортеж из объектов, образованных из значений аргументов, является элементом множества кортежей в расширении отношения, обозначенного константой отношения в интерпретации «I»:

$$\tau((Rt_1 K t_n)) = \begin{cases} \text{true cc } (\sigma(t_1), K, \sigma(t_n) \in \text{ext } (\sigma(R)) \text{ или} \\ \text{false} \end{cases}$$

Примечание — Атомарное высказывание формы (R) будет истинным только в случае пустого предложения $\langle\ \rangle \in$ ext (σ (R)).

B.3.3.3 Нет (not)

Отрицание является истинным только в том случае, если отрицательное предложение является ложным (false) в интерпретации «I»:

$$\tau((\text{not } \varphi)) = \begin{cases} \text{true } \tau(\varphi) = \text{false} \\ \text{в других случаях — false} \end{cases}$$

B.3.3.4 Или (or)

Дизъюнкция является истинной только в том случае, если по меньшей мере, один дизъюнкт является истинным в интерпретации «I»:

$$\tau((\text{or } \varphi_1 K \varphi_n)) = \begin{cases} \text{true } \tau(\varphi_j) = \text{true для некоторых значений } j,\ 0 \le j \le n \\ \text{в других случаях — false} \end{cases}$$

Примечание — Является истинным в любой интерпретации.

B.3.3.5 И (and)

Конъюнкция является истинной только в том случае, если конъюнкт является истинным в интерпретации «I»:

$$\tau((\text{and } \varphi_1 \mid \varphi_n)) = \begin{cases} \text{true } \tau(\varphi_j) = \text{true для всех значений } j,\ 0 \le j \le n \\ \text{в других случаях — false} \end{cases}$$

Примечание — Является истинным в любой интерпретации.

B.3.3.6 Заключает в себе (implies)

Если каждый antecedent в импликации является истинным, то импликация в целом является истинной в том случае, если consequent является истинным в интерпретации «I». Если любой из antecedents является ложным, то импликация в целом является истинной, независимо от истинного значения consequent в интерпретации «I»:

$$\tau((\text{implies } \varphi_1 K \varphi_n \varphi)) = \begin{cases} \text{true для некоторых значений } j,\ \tau(\varphi_j) = \text{false или } \tau(\varphi) = \text{true} \\ \text{в других случаях — false} \end{cases}$$

B.3.3.7 Если (iff)

Эквивалентность является истинной только в том случае, если все аргументы имеют одинаковое истинное значение в интерпретации «I»:

$$\tau((\text{iff } \varphi_1\ \varphi_2)) = \begin{cases} \text{true } \tau(\varphi_1) = \tau(\varphi_2) \\ \text{в других случаях — false} \end{cases}$$

B.3.3.8 Существует (exists)

Экзистенционально квантифицируемое предложение является истинным только в том случае, если вложенное предложение является истинным для некоторого значения переменных, указанных в аргументе «I»:

$$\tau((\text{exists } (\nu_1 \lessdot \nu_n)\ \varphi)) = \begin{cases} \text{true } \tau\ (\varphi\ ((\nu_1, ..., \nu_n))) = \text{true для некоторых значений I — variant} \\ \text{в других случаях — false} \end{cases}$$

B.3.3.9 Для всех (forall)

Универсально квантифицируемое предложение является истинным только в том случае, если вложенное предложение является истинным для каждого значения переменных, указанных в аргументе «I»:

$$\tau((\text{forall } (\nu_1 \lessdot \nu_2)) = \begin{cases} \text{true } \tau\ (\varphi\ ((\nu_1, ..., \nu_n))) = \text{true для каждого значения I — variant} \\ \text{в других случаях — false} \end{cases}$$

Приложение С
(справочное)

Пример описания процесса с использованием ядра PSL

Целью настоящего приложения является описание подробного сценария, в котором язык спецификации процесса, соответствующий комплексу стандартов ИСО 18629, используется в информационных целях и включает в себя многочисленные производственные функции.

Этот сценарий является сценарием производственной возможности «взаимодействия». Это означает, что его цель заключается в том, чтобы показать, как следует использовать PSL для облегчения информационного обмена знаниями о процессах производства. Этот сценарий сконцентрирован на обмене знаниями от плановых органов процесса до мастера, распределяющего работу в цехе.

Далее приведено описание производственных условий, в которых происходит вышеупомянутый обмен знаниями[1)]:

«Производственная фирма выпускает ряд изделий. Каждому изделию соответствуют точно установленный номер детали и один или более планов процесса. Один из планов процесса может быть более предпочтительным, но другие планы также являются действительными и обеспечивают выпуск приемлемых деталей. План процесса имеет список операций, которые должны быть выполнены в заданной последовательности. План процесса содержит определенную информацию о каждой операции: название (имя) операции, необходимые ресурсы, время, запланированное на выполнение операции, время настройки, различные параметры изделия и т. д.

В определенный момент времени фирма получает набор потребительских заказов. В каждом заказе указаны дата платежа и конкретное число некоторых деталей. Выполнение заказа означает завершение каждой операции по конкретному плану процесса для изготовления конкретного изделия.

Плановый производственный отдел должен составить график для цеха, в котором должен быть указан (для каждого потребительского заказа и каждой необходимой операции) обрабатывающий центр, на котором следует выполнять операцию, когда следует начинать и завершать операцию. График передают рабочим, которые проводят работы в соответствии с этим графиком».

Этот сценарий касается, в частности, обмена знанием о плане процесса, указанного выше. Полагают, что каждая производственная функция имеет собственное «родственное» представление, которое подходит к определенным инструментам, специализированным для этой функции (т. е. плановик процесса использует инструмент планирования процесса, а составитель графика — инструмент оперативного управления производством). Очевидно, что возможность взаимодействия является тривиальным актом, если совместные участники «породственному» используют общий язык/представление. Важным аспектом этой связи является сложный перевод (трансляция).

Взаимодействие для содействия этой связи обеспечивают следующим образом:

1) представление плана процесса переводят на язык спецификации процесса (PSL) и

2) представление PSL транслируется (переводится) в представление графика оперативного управления производством.

Транслятор можно написать между двумя представлениями (источником и целью). Недостатком является увеличение числа необходимых трансляторов с O(n) до $O(n^2)$ (где n равно числу трансляторов), если требуется перевод с большим числом представлений (например, между плановиком и цеховой системой управления).

С.1 Представления источника и цели

Представлениями, использованными в настоящем примере, являются IDEF3 [9] и специальный планировщик программ, используемые для планирования процесса и составления графика соответственно.

IDEF3 специально создан для сбора описаний последовательностей действий и является широко известным представлением знаний о процессе.

Планировщик является дополнением коммерческой библиотеки прикладных программ, предназначенных для составления графиков конечного объема.

С.2 Завод Camile Motor Works

Изделие, процесс и заводские знания, представленные в этом сценарии, взяты из примера, представляющего конкретный случай планирования и управления. Цель — дать подробный отчет по виртуальному моторному заводу, Camile Motor Works (CMW). Каждая характеристика и концепция, подтвержденные документами, базируются на факторах, относящихся к разряду критических для принятия решения (упреждающего или реагирующего) на одном или нескольких реальных заводах.

Этот случай включает в себя нормальное производственное планирование и управленческую деятельность с момента приема рабочего заказа, включая срок ведения переговоров, до момента, когда готовые изде-

---

[1)] Данный сценарий соответствует представленному Джефри Германном в группу разработки сценария на языке PSL, Университет штата Мериленд, Колледж Парк.

лия поступают на склад или для отправки. Завод CMW характеризуется как предприятие, выпускающее серию автомобилей масштабной модели, сконструированных с использованием:

- ряда закупаемых деталей, применяемых во время сборки;
- закупаемых деталей, проходящих дополнительную обработку на заводе (сборка, окраска и т. д.);
- собственных компонентов, изготовленных на заводе из сырья (различные корпуса, таблички, блок переключения передач, рамы, блок двигателя, проводка электропитания, шины, обода и др.).

Завод CMW может обеспечивать выполнение большого числа разных производственных процессов, поэтому является гибридной средой.

На заводе имеется восемь основных цехов (мастерских, отделов), составляющих 43 первичных ресурса. Завод нанимает примерно 50 человек для управления заводом и поддержания ресурсов.

С.3 Изделие: GT-350

В описании завода представлены три изделия (GT-200, GT-250, GT-350). Этого достаточно для обеспечения возможности взаимодействия программ с целью ограничения области применения и обращения к наиболее сложному изделию. Изделие GT-350 характеризуется как технологически сложный подвижный автомобиль масштабной модели, который продают с помощью маркетинга, когда информацию об образце продукции посылают по почте в исполнительные органы компаний с вариантами кодовых номеров разных типов изделий для выбора. Изделие может быть персонализировано по выбору покупателя двумя — шестью символами на идентификационной табличке (ID-пластине). Модель выпускают красного или белого цвета (элементы питания не включают). Это изделие высокой стоимости не передают на склад для хранения, а изготовление начинают после получения заказа. Поставку выполняют в течение 4—6 недель, предварительно оговорив сроки с покупателем.

Конструкция изделия GT-350 приведена в таблице С.1. Отступы в таблице используют для указания структуры компонентов комплектующих деталей. Каждый компонент имеет однозначно определяемый номер детали. Способ производства каждого компонента указан в последней правой графе. Производство некоторых деталей осуществляют на самом заводе или детали собирают из комплектующих частей, которые покупают или поставляют по контрактам с другими производителями.

Таблица С.1 — Конструкция изделия GT-350

Наименование компонента			Количество	Номер компонента	Способ производства
Шасси			1	350-ШАССИ	Сборка
	Двери		2	X50-ДВЕРИ	Закупка
	Цельный корпус		1	X50-КОРПУС	Заводское изготовление
	Рама		1	350-РАМА	Заводское изготовление
Двигатель			1	350-ДВИГАТЕЛЬ	Сборка
	Блок		1	350-БЛОК	Заводское изготовление
	Опора		1	350-ОПОРА	Заводское изготовление
	Электропроводка		1	350-ЖГУТ	Заводское изготовление
		Часть провода	12	350-ПРОВОД	Заводское изготовление
		Часть провода	1	X50-ПРОВОД	Заводское изготовление
Привод			1	350-ПРИВОД	Сборка
	Мотор		1	350-МОТОР	Закупка
	Электроника		1	350-PCB	Заводское изготовление
Салон			1	350-САЛОН	Сборка
	Кабина		1	350-КАБИНА	По контракту
		Приборная панель	1	350-ПАНЕЛЬ	Закупка
		Фары	2	350-ФАРЫ	Закупка
	Сиденья		1	350-СИДЕНЬЯ	Закупка
	Переключение передачи		1	X50-ПЕРЕДАЧА	Заводское изготовление

Окончание таблицы С.1

Наименование компонента			Количество	Номер компонента	Способ производства
Отделка			1	350-ОТДЕЛКА	Сборка
	По заказу		1	350-ЗАКАЗ	Сборка
	Номерные знаки		2	350-№ ЗНАКИ	Заводское изготовление
	Колеса		4	350-КОЛЕСА	Сборка
		Шины	1	350-ШИНЫ	Заводское изготовление
		Ободы	1	350-ОБОДЫ	Заводское изготовление
	Бирки		1	350-БИРКИ	Закупка
	ID-пластина		1	350-ID ПЛАСТИНА	Заводское изготовление
	Подставка		1	350-ПОДСТАВКА	Заводское изготовление

С.4 Отделы

На заводе CMW имеются указанные далее цеха (подробное описание приведено в [9]).

С.4.1 Литейный цех

Литейный цех производит фасонные металлические детали из металлических заготовок путем плавки заготовки и отливки частей.

С.4.2 Работа с резиной

В процессе работы с резиной изготавливают шины (350-шины) и детали блока переключения передач (х50-передача). Шины изготавливают из резиновой смеси путем штамповки в пресс-форме и распиловки отдельных шин из пресс-формы. Блок переключения передач изготавливают из резиновой смеси и отливки деталей. Перед началом отливки изложницы нагревают.

С.4.3 Листовой металл

Цех обработки листового металла производит ряд деталей для изделия GT-350 из поставляемого тонколистового металла. К ним относятся следующие части: 350-номерные знаки, х50-корпус, 350-ID пластина и 350-рама без окраски.

С.4.4 Механический цех

Механический цех обеспечивает инструментальную оснастку для изготовления двух деталей для изделия GT-350, используя покупное исходное сырье и запас заготовок литейного цеха. Этот цех выпускает следующие детали: 350-блок и 350-подставку без окраски.

С.4.5 Провод и кабель

Цех, изготавливающий провода и кабеля, выпускает 350-жгут и 350-набор проводов из 12 проволок мерной длины (350-провод). Кабель, используемый в жгуте, и проволочный набор изготавливают из металлических прутков, которые вытягивают в проволоку мерной длины тонкого сечения. Проволоку скручивают и покрывают цветной оболочкой.

С.4.6 Электроника

Цех электроники выпускает модули 350-PCB, которые управляют функциями модели 350. Здесь также изготавливают модули для другой модели. Компоненты 350-PCB сначала собирают на плате, которую экранируют. Некоторые компоненты устанавливают автоматически, остальные — вручную. Платы запаивают и проверяют на устойчивость к воздействию механического напряжения. Функциональной проверке подвергают ряд изделий из каждой партии. В зависимости от результатов проверки партия может быть принята, забракована или отправлена на переработку. Платы переделывают не более одного раза.

С.4.7 Окраска

Цех окраски обеспечивает окраску в заданный цвет двух деталей изделия GT-350, используя запас деталей, изготовленных в механическом цехе и цехе обработки листового металла. Окрашенными деталями являются 350-рама и 350-подставка.

С.4.8 Окончательная сборка

Сборочный цех обеспечивает полную интеграцию составных компонентов изделия. Здесь также осуществляют некоторые модификации конечного изделия (например, гравировку). Этот цех выпускает готовое изделие GT-350.

С.5 Процессы производства GT-350

В этом разделе приведено общее описание процессов работы разных цехов, обеспечивающих совокупность действий высокого уровня, которые предпринимаются для создания изделия GT-350. Как следует из конструкции изделия GT-350 (см. таблицу С.1), комплектующие компоненты этого изделия либо закупают, либо произ-

водят по контракту с поставщиками, либо изготавливают на заводе. Описание этих процессов сводится к действиям, которые выполняют для производства комплектующих компонентов на заводе. Этот обзор производственного процесса дает всестороннюю картину действия «изготовить GT-350», которое расширяется по нисходящей до подробных цеховых уровней.

В соответствии с рисунком С.1 процесс производства GT-350 подразделен на шесть основных видов работ. Первые пять (изготовить кабину (make interior), изготовить привод (make drive), выполнить отделку (make trim), изготовить двигатель (make engine) и изготовить шасси (make chassis)) являются неупорядоченными по отношению друг к другу, но они должны быть завершены до окончательной сборки (final assembly).

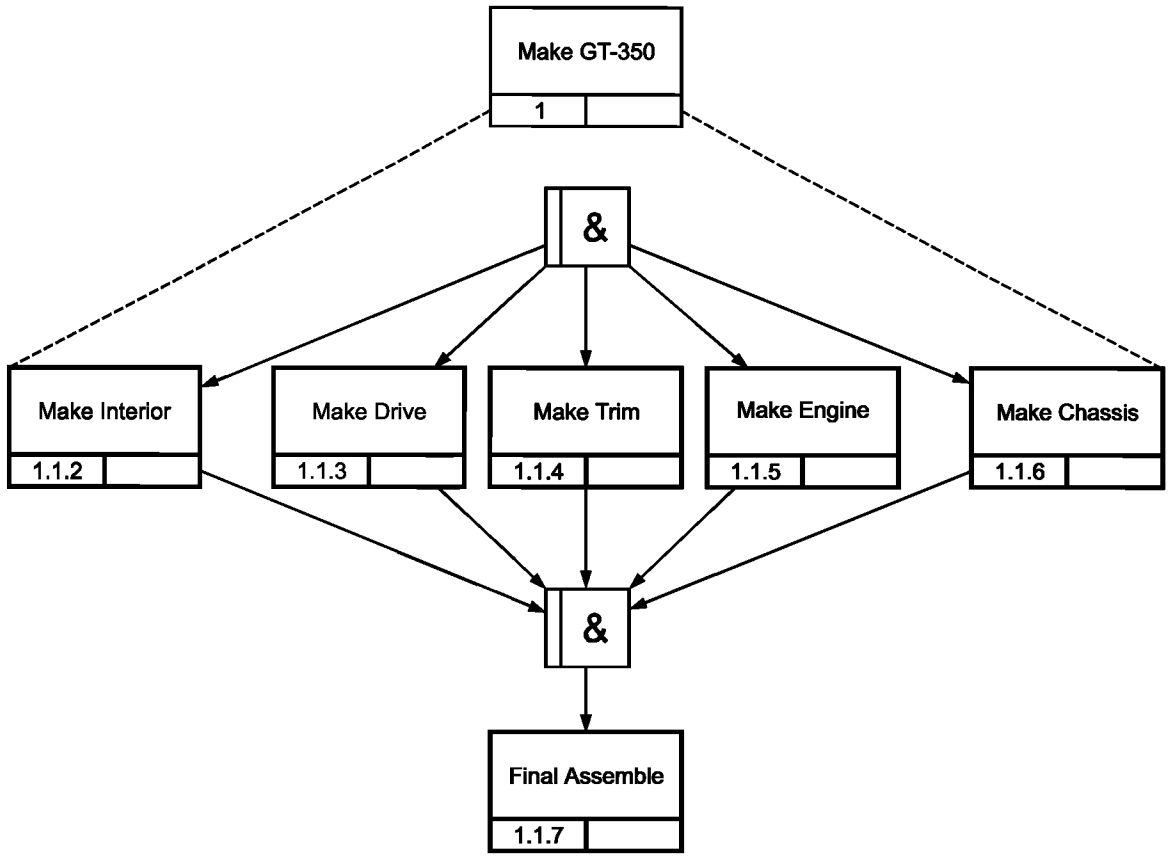

Рисунок С.1 — Процесс верхнего уровня производства изделия GT350 [9]

Представление процесса, указанного на рисунке С.1, с помощью ядра языка спецификации процесса:

(activity make-chassis)

(activity make-interior)

(activity make-drive)

(activity make-trim)

(activity make-engine)

(activity final-assembly)

Каждое из этих абстрактных действий может быть детализировано, однако для данного примера нет такой необходимости.

На основе представления IDEF3 абстрактных действий (в терминах представления процесса), встречающихся на разных стадиях производства, можно выделить несколько примеров описаний процессов с помощью представления ядра PSL по настоящему стандарту.

С.5.1 Абстрактное действие «изготовление двигателя»

Двигатель 350-Engine изготавливают в процессе работ, выполняемых в нескольких цехах завода CMW. Производственный процесс показан на рисунке С.2. Двигатель комплектуют из блока make block, комплекта кабелей make harness и разводки проводов make wires. Описание вспомогательных процессов приведено в С.5.2. Двигатель 350-Engine собирают на стенде А004; на установку одной детали требуется 5 мин.

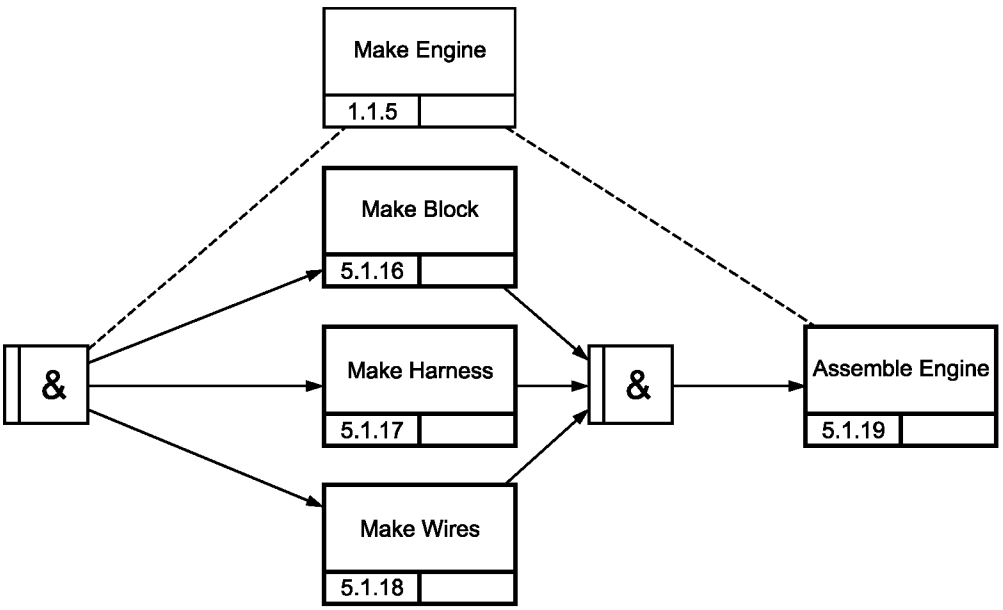

Рисунок С.2 — Процессы изготовления двигателя 350-Engine

Представление некоторых действий процесса на основе ядра PSL и информации, имеющей отношение к этому процессу, на стадии изготовления двигателя (make-engine) имеет следующий вид.

Примечание — Последний элемент в предложении формата обмена знанием (KIF) не входит в ядро PSL, однако его оставляют с целью сохранения общей согласованности определенного выражения.

```
(and
    (forall ?occ)
        (implies (and  (activity_occurrence ?occ)
                       (occurrence-of ?occ assemble-engine))
                 (exists ?occ2)
                       (and (activity_occurrence ?occ2)
                       (occurrence-of ?occ2 make-gt350-proc))
```

С.5.2 Вспомогательные процессы «изготовить двигатель»

В настоящем подразделе приведено описание вспомогательных процессов.

С.5.2.1 Изготовить блок

Блок 350-Block изготавливают как часть, предназначенную для сборки двигателя 350-Engine. Сборка включает в себя совместную работу литейного и механического цехов согласно рисунку С.3. В литейном цехе производят отливку металлической части (Produce Molded Metal Item), а в механическом — машинную обработку блока (Machine Block).

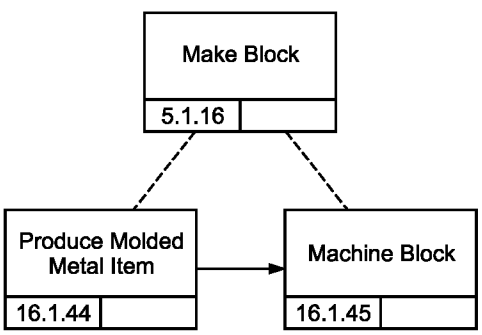

Рисунок С.3 — Процесс производства блока 350-Block [9]

С.5.2.2 Изготовить жгут электропроводки (Make Harness)

Изготовление и разводку проводов электропитания (350-Harness) выполняют на этапе сборки двигателя 350-Engine (рисунок С.4) в цехе по изготовлению жгутов и проводки. На рисунке С.5 приведен процесс производства электропроводки. Разводку электропроводки выполняет электрик на специальном стенде. На разводку комплекта проводов требуется 10 мин.

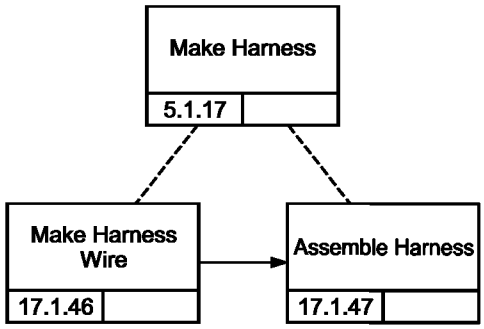

Рисунок С.4 — Процесс изготовления жгута 350-Harness [9]

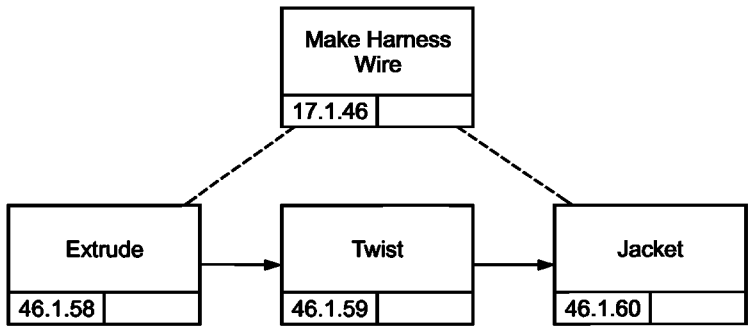

Рисунок С.5 — Процесс изготовления электропроводки 350-Wire [9]

С.5.2.3 Изготовление проволоки

Комплект проводов изготавливают на этапе сборки двигателя 350-Engine в цехе по изготовлению жгутов, кабелей и электропроводов. Расширенный процесс производства комплекта электропроводки изображен на рисунке С.6. Комплект электропроводки собирает электрик на специальном стенде. На сборку комплекта требуется 5 мин.

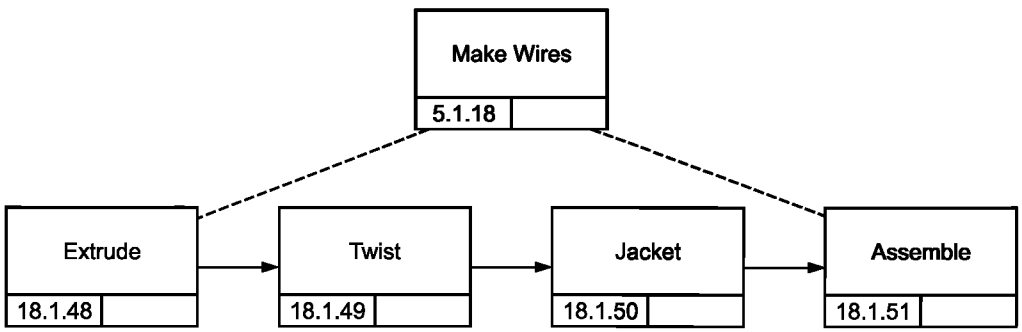

Рисунок С.6 — Процесс изготовления проволоки x50-Wire для электропроводки

Представление на основе ядра PSL некоторых действий и информации, имеющей отношение к процессу изготовления электропроводки, имеет следующий вид:

```
(forall (?occ1)
        (implies (occurrence_of ?occ assemble_engine)
                 (exists (?occ2 ?occ3 ?occ4 ?x1 ?x2 ?t1 ?t2)
                         (and (occurrence_of ?occ2 make_chassis)
                              (occurrence_of ?occ3 machine_block)
                              (occurrence_of ?occ4 final_assembly)
                              (participates_in ?x1 ?occ2 ?t1)
                              (participates_in ??occ ?t1)
                              (participates_in ?x2 ?occ ?t2)
                              (chassis ?x1)
                              (engine ?x2)
                              (before (beginof ?occ2) (beginof ?occ3))
                              (before (endof ?occ3) (beginof ?occ4)))))
```

Приложение D
(справочное)

Условные обозначения формы Бэкуса-Наура

Используют следующие условные обозначения расширенной формы Бэкуса-Наура (BNF):

- Вертикальная черта «|» указывает исключающую дизъюнкцию, например, если С1 и С2 являются двумя синтаксическими категориями, то «С1|С2» указывает на появление либо элемента С1, либо элемента С2, но не на появление обоих элементов. Отсутствие черты между двумя элементами указывает на конкатенацию.

Звездочка «*» сразу за логической структурой указывает, что может быть любое конечное число (включая 0) элементов логической структуры.

- Верхний индекс в виде знака «+» сразу за логической структурой указывает, что может быть один или больше экземпляров логической структуры.

- Фигурные скобки «{» и «}» используют для обозначения группирования. Так «{С1 | С2}+» указывает на наличие одного или более элементов либо С1, либо С2.

- Логическая структура, взятая в квадратные скобки, например «[С1 | С2]», указывает, что элемент указанной логической структуры является необязательным.

- Нетерминальный символ, представляющий категории выражений, начинается знаком «<» и заканчивается знаком «>». Например, «<b-var> ::= ?<b-indcon>» указывает, что переменная величина должна начинаться с вопросительного знака.

- При необходимости символ пробела представляют выражением «<space>».

Приложение ДА
(справочное)

Сведения о соответствии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации

Таблица ДА.1

Обозначение ссылочного международного стандарта	Степень соответствия	Обозначение и наименование соответствующего национального стандарта
ИСО/МЭК 8824-1	IDT	ГОСТ Р ИСО/МЭК 8824-1—2001 «Информационная технология. Абстрактная синтаксическая нотация версии один (АСН.1). Часть 1. Спецификация основной нотации»
ИСО10303-1	IDT	ГОСТ Р ИСО 10303-1—99 «Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 1. Общие представления и основополагающие принципы»
ИСО15531-1	IDT	ГОСТ Р ИСО 15531-1—2008 «Промышленные автоматизированные системы и интеграция. Данные по управлению промышленным производством. Часть 1. Общий обзор»
ИСО15531-42:2005	IDT	ГОСТ Р ИСО 15531-42—2010 «Системы промышленной автоматизации и интеграция. Управляющая информация промышленным производством. Управление использованием ресурсов. Часть 42. Модель данных времени использования ресурсов»

Примечание — В настоящей таблице использовано следующее условное обозначение степени соответствия стандартов:

- IDT — идентичные стандарты.

Библиография

[1] ИСО 10303-41 Системы промышленной автоматизации и интеграция. Представление данных о продукции и обмен данными. Часть 41. Интегрированные родовые ресурсы. Основы описания и поддержки продукции
(ISO 10303-41) (Industrial automation systems and integration — Product data representation exchange — Part 41: Integrated generic resource: Fundamentals of product description and support)

[2] ИСО 13584-1 Системы промышленной автоматизации и интеграция. Библиотека данных на детали. Часть 1. Обзор и основные принципы
(ISO 13584-1) (Industrial automation systems and integration — Parts library — Part 1: Overview and fundamental principles)

[3] ИСО 15926-1 Системы промышленной автоматизации и интеграция. Интеграция данных о сроке службы нефтехимических установок, включая установки по добыче нефти и газа. Часть 1. Общее представление и основные принципы
(ISO 15926-1) (Industrial automation systems and integration — Integration of life-cycle data for process plants including oil and gas production facilities — Part 1: Overview and fundamental principles)

[4] Allen J., Hayes P. Моменты и точки временной логики на основе интервала. Технический доклад (TR 180). Отделы компьютерной науки и философии, университет г. Рочестер, 1987 г.
(Allen J., Hayes P. Moments and points in an interval-based temporal logic, Technical Report (TR 180), Departments of Computer Science and Philosophy, University of Rochester, 1987)

[5] Barwise J., Etchemendy J. Язык логики первого порядка, 3-е издание. Примечания № 34 к лекции CSLI, 1992 г.
(Barwise J., Etchemendy J. The language of first-order logic, 3rd Edition, CSLI Lecture Notes № 34, 1992)

[6] Cann R. Формальная семантика, введение. Печать университета г. Кембридж, 1994 г.
(Cann R. Formal semantics, an introduction, Cambridge University Press, 1994)

[7] Genesereth M., Fikes R. Формат обмена знаниями (версия 3.0). Справочник. Отдел компьютерной науки, университета г. Стенфорд. 1992 г.
(Genesereth M., Fikes R. Knowledge Interchange Format (Version 3.0) — Reference Manual, Computer Science Dept., Stanford University, Stanford, CA., 1992)

[8] Формат обмена знаниями. Часть 1. Ядро KIF, ISO/JTC1/SC32/WG2,WD, 1999 г.
(Knowledge Interchange Format, Part 1: KIF-Core, ISO/JTC1/SC32/WG2, WD, 1999)

[9] Публикация 184 по федеральным стандартам обработки информации, определение интеграции для информационного моделирования (IDEF3), FIPS PUB 184, Национальный институт стандартов и технологии, декабрь 1993 г. IDEF3. Доступ в сети Интернет: gost.e-ecolog.ru
(Federal Information Processing Standards Publication 184, Integration Definition for Information Modeling (IDEF3), FIPS PUB 184, National Institute of Standards and Technology, December 1993. IDEF3. Available from the Internet: <gost.e-ecolog.ru >)

[10] Mayer R.J., Menzel C.P., Painter M.K., de Witte P.S., Blinn T., Perekath B. Интеграция информации для параллельного проектирования (IICE). Сообщение о методе сбора описаний процесса IDEF3, KBSI Inc. AL-TR-1995, 1995 г.
(Mayer R.J., Menzel C.P., Painter M.K., de Witte P.S., Blinn T., Perakath B. Information integration for concurrent engineering (IICE) IDEF3 process description capture method report, KBSI Inc, AL-TR-1995, 1995)

[11] Stephan T. Polyak и Stuart Aitken Сценарии возможности взаимодействия процесса производства, проект NIST-MSID-SIMA, AIAI-PR-68, 1988 г.
(Stephen T. Polyak and Stuart Aitken Manufacturing Process Interoperability Scenario, NISTMSID-SIMA Project, AIAI-PR-68, 1998)

[12] Uschold M. и Gruninger M. Онтологии. Принципы, методы и приложения. Обзор инженерных знаний, Том 11, стр. 96—137, 1996 г.
(Uschold M. and Gruninger M. Ontologies: Principles, Methods, and Applications, Knowledge Engineering Review, Vol. 11, pp. 96—137, 1996)

[13] Van Benthem J. Логика времени. Дордрехт. Голландия, D. Reidel Pub Co., 1983 г.
(Van Benthem J. The logic of time, Dordrecht, Holland, D. Reidel Pub. Co., 1983)

УДК 65.011:56.681.3 ОКС 25.040.40 Т58

Ключевые слова: автоматизированные промышленные системы, интеграция, жизненный цикл систем, управление производством

Редактор *Т. А. Леонова*
Технический редактор *В. Н. Прусакова*
Корректор *Л. Я. Митрофанова*
Компьютерная верстка *А. П. Финогеновой*

Сдано в набор 20.02.2014. Подписано в печать 10.04.2014. Формат $60\times84^{1}/_{8}$. Бумага офсетная. Гарнитура Ариал. Печать офсетная. Усл. печ. л. 3,72. Уч.-изд. л. 3,20. Тираж 69 экз. Зак. 344.

ФГУП «СТАНДАРТИНФОРМ», 123995 Москва, Гранатный пер., 4.
gost.e-ecolog.ru gost.e-ecolog.ru
Набрано и отпечатано в Калужской типографии стандартов, 248021 Калуга, ул. Московская, 256.

gost.e-ecolog.ru