ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 60.0.0.8— 2023

Роботы и робототехнические устройства

ОНТОЛОГИИ РОБОТОТЕХНИКИ

Общие положения, основные понятия, термины и определения

Издание официальное

Москва
Российский институт стандартизации
2023

gost.e-ecolog.ru

Предисловие

1 ПОДГОТОВЛЕН Федеральным государственным автономным научным учреждением «Центральный научно-исследовательский и опытно-конструкторский институт робототехники и технической кибернетики» (ЦНИИ РТК) на основе собственного перевода на русский язык англоязычной версии стандарта, указанного в пункте 4

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 141 «Робототехника»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 15 сентября 2023 г. № 850-ст

4 Настоящий стандарт является модифицированным по отношению к стандарту IEEE Std 1872—2015 «Стандартные онтологии IEEE для робототехники и автоматизации» (IEEE Std 1872—2015 «IEEE standard ontologies for robotics and automation», MOD) путем внесения технических отклонений, объяснение которых приведено во введении к настоящему стандарту.

Наименование настоящего стандарта изменено относительно наименования указанного стандарта для приведения в соответствие с ГОСТ Р 1.5—2012 (пункт 3.5) и для увязки с наименованиями, принятыми в существующем комплексе национальных стандартов Российской Федерации

5 ВВЕДЕН ВПЕРВЫЕ

*Правила применения настоящего стандарта установлены в статье 26 Федерального закона от 29 июня 2015 г. № 162-ФЗ «О стандартизации в Российской Федерации». 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.e-ecolog.ru

© Оформление. ФГБУ «Институт стандартизации», 2023

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 и определения 2
4 Сокращения и обозначения 5
5 Онтология SUMO 5
6 Аксиомы CORAX 9
7 Аксиомы CORA 16
8 Аксиомы RPARTS 23
9 Аксиомы POS 24

Введение

Требования стандартов комплекса ГОСТ Р 60 распространяются на роботы и робототехнические устройства. Целью стандартов является повышение интероперабельности роботов и их компонентов, а также снижение затрат на их разработку, производство и обслуживание за счет стандартизации и унификации процессов, интерфейсов, узлов и параметров.

Стандарты комплекса ГОСТ Р 60 представляют собой совокупность отдельно издаваемых стандартов. Стандарты данного комплекса относятся к одной из следующих тематических групп: «Общие положения, основные понятия, термины и определения», «Технические и эксплуатационные характеристики», «Безопасность», «Виды и методы испытаний», «Механические интерфейсы», «Электрические интерфейсы», «Коммуникационные интерфейсы», «Методы моделирования и программирования», «Методы построения траектории движения (навигация)», «Конструктивные элементы». Стандарты любой тематической группы могут относиться как ко всем роботам и робототехническим устройствам, так и к отдельным группам объектов стандартизации — промышленным роботам в целом, промышленным манипуляционным роботам, промышленным транспортным роботам, сервисным роботам в целом, сервисным манипуляционным роботам, сервисным мобильным роботам, а также к морским робототехническим комплексам.

Настоящий стандарт относится к тематической группе «Общие положения, основные понятия, термины и определения» и относится ко всем роботам и робототехническим устройствам.

Настоящий стандарт является модифицированным по отношению к стандарту IEEE Std 1872—2015, разработанному Институтом инженеров по электротехнике и радиоэлектронике (Institute of electrical and electronics engineers, incorporated, IEEE) при поддержке постоянного комитета по стандартизации общества робототехники и автоматизации IEEE, в соответствии с принципами стандартизации, установленными в Решении о принципах разработки международных стандартов, руководств и рекомендаций Комитета по техническим барьерам в торговле Всемирной торговой организации, для его приведения в соответствие с требованиями основополагающих национальных и межгосударственных стандартов.

В настоящий стандарт внесены следующие технические отклонения по отношению к IEEE Std 1872—2015:

- исключены примечания и сноски примененного стандарта, которые нецелесообразно применять в национальной стандартизации в связи с их содержанием, имеющим справочный характер и относящимся к системе стандартизации США;
- раздел 1 «Область применения» приведен в соответствие с требованиями ГОСТ Р 1.5—2012 с исключением смысловых повторов: пункты 1.1 и 1.2 примененного стандарта объединены в пункт 1.1;
- из раздела 2 «Нормативные ссылки» исключены ссылки на публикации, не являющиеся стандартами и не имеющие аналогов среди межгосударственных и национальных стандартов, в соответствии с требованиями ГОСТ Р 1.5—2012;
- в раздел 2 добавлена нормативная ссылка на ГОСТ Р ИСО/МЭК 21838-1—2021;
- раздел 3 «Определения, сокращения и акронимы» и раздел 4 «Условные обозначения» примененного стандарта переработаны в раздел 3 «Термины и определения» и раздел 4 «Обозначения и сокращения» в соответствии с требованиями ГОСТ 1.5—2001, ГОСТ Р 1.5—2012 и ГОСТ Р 1.7—2014;
- в разделе 3 термины и их определения приведены в соответствии с идентичными национальными стандартами;
- в стандарте приведены русскоязычные эквиваленты стандартизованных терминов на английском языке;
- терминологические статьи расположены в алфавитном порядке букв русского языка для обеспечения соответствия требованиям ГОСТ Р 1.5—2012;
- исключены примечания типа «смотри также» и «противоположные по смыслу термины», которые не несут смысловой нагрузки и определяются соответствующими терминологическими статьями;
- в разделе 5 введена нумерация пунктов и подпунктов в соответствии с требованиями ГОСТ Р 1.5—2012 и ГОСТ Р 1.7—2014 для обеспечения возможности ссылок и цитирования терминов в других стандартах;
- формулы SUO-KIF заменены OWL-кодами онтологий в соответствии с требованиями ГОСТ Р ИСО/МЭК 21838-1—2021;

- в разделах 8 и 9 исключены некоторые формулы SUO-KIF, выражающие частные примеры, которые невозможно записать стандартными OWL-кодами онтологий платформы Protégé (требуется программирование на языке Python или C++);

- исключены справочные приложения (приложение А, приложение В и приложение С) как не имеющие значения для данного стандарта;

- ключевые слова приведены в библиографических данных в соответствии с требованиями ГОСТ 1.5—2001.

Заключенная в круглые скобки часть термина может быть опущена при использовании термина в документах по стандартизации.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Роботы и робототехнические устройства

ОНТОЛОГИИ РОБОТОТЕХНИКИ

Общие положения, основные понятия, термины и определения

Robots and robotic devices.
Ontologies for robotics.
General provisions, fundamental concepts, terms and definitions

Дата введения — 2024—01—01

1 Область применения

Настоящий стандарт определяет основные онтологии, которые обеспечивают возможность представлять, обосновывать и передавать знания в области робототехники и автоматизации. Стандарт обеспечивает унифицированный способ представления знаний и предоставляет общий набор определений и терминов, позволяющий осуществлять однозначную передачу знаний между любой группой людей, роботов и других искусственных систем.

Настоящий стандарт определяет основную онтологию для робототехники и автоматизации CORA (Core Ontology for Robotics and Automation) с представлением фундаментальных понятий, из которых строятся более подробные концепции, принадлежащие другим онтологиям в области робототехники. Настоящий стандарт также определяет методологию разработки онтологий, используемую для построения онтологий ORA (Ontology for Robotics and Automation). Информация, представленная в настоящем стандарте, охватывает аппаратное и программное обеспечение роботов, действия и цели, окружающую среду, причины и следствия выполнения действий, а также взаимоотношения между роботами и людьми, но не ограничивается этим.

Основными пользователями настоящего стандарта являются производители роботов, системные интеграторы, конечные пользователи роботов (производители деталей, автомобильная промышленность, строительная промышленность, поставщики услуг и решений и т. д.), поставщики оборудования для роботов, разработчики программного обеспечения для роботов, исследователи и разработчики.

2 Нормативные ссылки

В настоящем стандарте использована нормативная ссылка на следующий стандарт:

ГОСТ Р ИСО/МЭК 21838-1—2021 Информационные технологии. Онтологии высшего уровня (TLO). Часть 1. Требования

Примечание — При пользовании настоящим стандартом целесообразно проверить действие ссылочных стандартов в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет или по ежегодному информационному указателю «Национальные стандарты», который опубликован по состоянию на 1 января текущего года, и по выпускам ежемесячного информационного указателя «Национальные стандарты» за текущий год. Если заменен ссылочный стандарт, на который дана недатированная ссылка, то рекомендуется использовать действующую версию этого стандарта с учетом всех внесенных в данную версию изменений. Если заменен ссылочный стандарт, на который дана датированная ссылка, то рекомендуется использовать версию этого стандарта с указанным выше годом утверждения (принятия). Если после утверждения настоящего стандарта в ссылочный стандарт, на который дана

Издание официальное

датированная ссылка, внесено изменение, затрагивающее положение, на которое дана ссылка, то это положение рекомендуется применять без учета данного изменения. Если ссылочный стандарт отменен без замены, то положение, в котором дана ссылка на него, рекомендуется применять в части, не затрагивающей эту ссылку.

3 Термины и определения

В настоящем стандарте применены следующие термины с соответствующими определениями:

3.1 **автоматический робот** (automated robot): Робот, выполняющий заданную задачу как автомат, не приспосабливаясь к изменениям окружающей среды и/или следуя сценариям.

3.2 **взаимодействие** (interaction): Процесс [*Процесс* (*Process*) в SUMO], в котором участвуют два агента [*Агент* (*Agent*) в SUMO], состоящий из двух подпроцессов, определяющих действие и реакцию.

Примечание — Подпроцесс действия, инициируемый агентом X для реципиента агента Y, вызывает подпроцесс реакции, имеющий Y в качестве агента и X в качестве реципиента.

3.3 **внешняя среда робота** (robotic environment): Физическая среда, в которой функционирует робототехнический комплекс.

3.4 **глобальная система координат** (global coordinate system): Условная система координат, выбранная агентом в качестве системы отсчета, которая составляет основную систему координат для этого агента.

Примечание — В иерархии локальных систем координат глобальная система координат является корнем дерева локальных систем координат.

3.5 **группа роботов** (robot group): Группа [*Группа* (*Group*) в SUMO] роботов, организованная для достижения по крайней мере одной общей цели.

3.6 **дистанционно управляемый робот** (remote-controlled robot): Робот, выполняющий задание под постоянным управлением человека-оператора, находящегося в удалении от робота и руководствующегося только своими непосредственными наблюдениями.

Примечание — В данном режиме робот не проявляет никакой инициативы и полагается на непрерывный или почти непрерывный ввод данных от человека-оператора.

3.7 **значение ориентации** (orientation value): Значение в системе координат, обозначающее конкретную ориентацию.

Примечание — Значения ориентации в одной системе координат могут быть сопоставлены с другими системами координат. Примером использования значения ориентации является выражение «робот ориентирован на 54° относительно опорного объекта».

3.8 **интерфейс робота** (robot interface): Устройство [*Устройство* (*Device*) в SUMO], состоящее из сенсорных, исполнительных и коммуникационных компонентов робота.

Примечание — Через интерфейс робот может воспринимать окружающую среду и воздействовать на нее, а также коммуницировать с другими агентами. Таким образом, интерфейс робота можно рассматривать как способ обозначения всех устройств, которые позволяют роботу взаимодействовать с окружающим миром. Каждый робот имеет только один интерфейс.

3.9 **искусственная система** (artificial system): Артефакт [*Артефакт* (*Artifact*) в SUMO], сформированный разными взаимодействующими устройствами [*Устройство* (*Device*) в SUMO] и другими объектами [*Объект* (*Object*) в SUMO] для выполнения некоторой функции.

3.10 **исполнительный компонент робота** (robot actuating part): Устройство [*Устройство* (*Device*) в SUMO], позволяющее роботу двигаться и действовать в окружающей среде.

3.11 **коллективный робототехнический комплекс** (collective robotic system): Робототехнический комплекс, в состав которого входит группа роботов.

3.12 **коммуникационный компонент робота** (robot communicating part): Устройство [*Устройство* (*Device*) в SUMO], служащее инструментом в процессе коммуникации робота с роботом или человека с роботом, позволяя роботу посылать информацию другому роботу или человеку (или принимать информацию от них).

3.13 **компонент робота** (robot part): Любое устройство [*Устройство* (*Device*) в SUMO], установленное на роботе и обеспечивающее функционирование робота.

Примечание — Устройства, рассматриваемые как компоненты робота, будучи установленными на нем, не всегда обязательно являются компонентами робота в онтологическом смысле, так как они существуют сами по себе и в большинстве случаев могут быть подсоединены к другим типам устройств. Например, источник питания по существу является устройством; однако конкретный экземпляр источника питания может в динамике рассматриваться как компонент робота в течение определенного интервала времени, когда он подключен к роботу. Если некоторое устройство является компонентом робота, то его части также считаются компонентом этого робота.

3.14 **локальная система координат** (local coordinate system): Система координат, ограниченная иерархической структурой систем координат и не расположенная в корне данной структуры.

3.15 **область ориентации** (orientation region): Область или сектор ориентации относительно выбранного объекта [*Объект* (*Object*) в SUMO].

Примечание — Например, сектор «юг» компаса представляет область ориентации в одномерной круговой системе координат компаса. В конечном счете, области позиции и области ориентации обозначаются похожими словами. Например, можно сказать, что робот расположен в позиции на севере или обращен к северу. Первое относится к области позиции, т. е. к северной области данной страны; второе относится к области ориентации, т. е. к сектору ориентации вокруг севера на компасе.

3.16 **область позиции** (position region): Абстрактная область позиции в системе координат.

Примечание — Область позиции определена точками позиций в заданной системе координат. Она определяет позиции качественно, например «слева от», «перед», «сверху от» и т. д. Данные выражения определяют области расположения объектов относительно выбранного объекта. Область позиции всегда генерируется заданным пространственным оператором, применяемым к списку выбранных объектов.

3.17 **обрабатывающий компонент робота** (robot processing part): Устройство, позволяющее роботу обрабатывать информацию.

3.18 **обрабатывающее устройство** (processing device): Электрическое устройство [*Электрическое устройство* (*Electric Device*) в SUMO], являющееся инструментом в подклассе компьютерного процесса [*Компьютерный процесс* (*Computer Process*) в SUMO].

3.19 **одинарный робототехнический комплекс** (single robotic system): Робототехнический комплекс, состоящий только из одного робота и одного или нескольких вспомогательных устройств.

3.20 **полностью автономный робот** (fully autonomous robot): Робот, выполняющий задание без вмешательства человека, адаптируясь к условиям эксплуатации и окружающей среды.

3.21 **полуавтономный робот** (semi-autonomous robot): Робот, выполняющий задание способом, при котором робот и человек-оператор планируют и выполняют задание, требующее разных уровней взаимодействия с человеком.

3.22 **преобразование** (transformation): Процесс перевода значений из одной системы координат в другую систему координат.

3.23

пространственное расположение (pose): Комбинация позиции и ориентации в пространстве.

Примечания

1 Под пространственным расположением **манипулятора** (4.14) обычно понимают позицию и ориентацию **рабочего органа** (4.12) или **механического интерфейса** (4.11).

2 Пространственное расположение **мобильного робота** (4.15) может включать совокупность пространственных расположений **мобильной платформы** (4.16) и любого манипулятора, установленного на мобильной платформе, относительно **системы координат мобильной платформы** (5.12).

[*ГОСТ Р 60.0.0.4—2023, статья 5.5*]

Примечание — Пространственное расположение объекта — это описание любых измерений позиции и ориентации данного объекта.

3.24 **пространственный оператор** (spatial operator): Математическая функция, позволяющая отображать эталонные объекты [*Объект* (*Object*) в SUMO] на области в системе координат.

3.25

робот (robot): Программируемый исполнительный механизм, обладающий определенным уровнем автономности и предназначенный для выполнения перемещения, манипулирования или позиционирования.

Примечания

1 В состав робота входит система управления.

2 Примерами конструктивных разновидностей роботов являются манипулятор, мобильная платформа и носимый робот.

[*ГОСТ Р 60.0.0.4—2023, статья 3.1*]

Примечание — В контексте настоящего стандарта робот — это устройство-агент [*Агент* (*Agent*) и *Устройство* (*Device*) в SUMO] в широком смысле, предназначенное для осуществления действий в физическом мире с целью выполнения одной или нескольких задач. В некоторых случаях, действия робота могут быть подчинены действиям других агентов [*Агент* (*Agent*) в SUMO], таких как программные агенты (боты) или люди. Робот состоит из надлежащих механических и электронных компонентов. Роботы могут образовывать социальные группы, в которых они взаимодействуют для достижения общей цели. Робот (или группа роботов) может создавать робототехнические комплексы вместе со специальными средствами, предназначенными для облегчения их работы.

3.26 **робототехнический комплекс** (robot system): Искусственная система, образованная одним или несколькими роботами и, по меньшей мере, одним устройством [Устройство (Device) в SUMO], поддерживающим работу робота (роботов).

3.27 **сенсорный компонент робота** (robot sensing part): Измерительное устройство [*Измерительное устройство* (*Measuring Device*) в SUMO], позволяющее роботу получать информацию об окружающей среде.

3.28 **система координат** (coordinate system): Абстрактная сущность [*Абстракт* (*Abstract*) в SUMO], используемая для задания местоположения и ориентации относительно одного эталонного объекта [*Объект* (*Object*) в SUMO].

Примечание — Системы координат связаны между собой иерархиями (т. е. «деревьями»). Например, на локальную систему координат робота ссылается сам робот. Эталонный объект не обязательно является началом системы координат. Система координат определяет по крайней мере одно измерение, в котором точки получают значения своих координат. Точки в данной системе координат могут быть сопоставлены с другими системами координат посредством преобразования.

3.29 **телеуправляемый робот** (teleoperated robot): Робот, выполняющий задание способом, при котором человек-оператор, используя сенсорную обратную связь, непосредственно управляет исполнительными механизмами либо постоянно назначает последовательность целей, находясь в отдалении от робота.

Примечание — Телеуправляемый робот выполнит последнюю команду, как только оператор прекратит посылать команды, даже если эта команда сложная или отнимает много времени.

3.30 **технический облик** (design): Утверждение [*Утверждение* (*Proposition*) в SUMO], которое обобщает структуру одного или нескольких артефактов [*Артефакт* (*Artifact*) в SUMO].

Примечание — Технический облик используют для обобщения информации в таких контекстах, как промышленная робототехника. Технический облик отличается от чертежа; чертеж представляет собой конкретизацию технического облика.

3.31 **точка позиции** (position point): Точка позиции в системе координат, обозначающая точное указание позиции данного объекта.

Примечание — Точки позиций всегда задаются в одной системе координат.

3.32 **физическая среда** (physical environment): Объект [*Объект* (*Object*) в SUMO], который имеет по крайней мере один конкретный компонент — область [*Область* (*Region*) в SUMO], в которой он расположен.

Примечание — Физическая среда относится по меньшей мере к одному эталонному объекту [*Объект* (*Object*) в SUMO], базирующемуся на области, в которой он определен.

3.33 **характеристика ориентации** (orientation measure): Характеристика [*Характеристика* (*Measure*) в SUMO], (физического) объекта [*Объект* (*Object*) в SUMO], содержащая информацию о направлении объекта относительно эталонного объекта системы координат ориентации.

3.34 **характеристика позиции** (position measure): Характеристика [*Характеристика* (*Measure*) в SUMO], (физического) объекта [*Объект* (*Object*) в SUMO], описывающая его позицию.

Примечание — Позиция может быть описана точкой или областью. Например, можно описать робота, как расположенного в координатах (x, y) в системе координат, либо перед ящиком, где «перед» означает направленную вперед коническую область, вершина которой расположена на ящике.

4 Сокращения и обозначения

4.1 В настоящем стандарте применены следующие сокращения:

ALFUS — уровни автономности для беспилотных систем (Autonomy Levels for Unmanned Systems)

CORA — основная онтология для робототехники и автоматизации (Core Ontology for Robotics and Automation)

CORAX — онтология, определяющая понятия и отношения, которые являются более общими, чем понятия, описанные в CORA, не учтенные онтологией SUMO (не является аббревиатурой как таковой)

ORA — онтология для робототехники и автоматизации (Ontology for Robotics and Automation)

ORA WG — рабочая группа ORA (ORA Working Group)

OWL — язык веб-онтологий (Web Ontology Language)

POS — онтология, которая отражает общие понятия о положении и ориентации (не является аббревиатурой как таковой)

RPARTS — компоненты робота (Robot Parts)

SUMO — онтология верхнего уровня (Suggested Upper Merged Ontology)

SUO-KIF — стандартный формат обмена знаниями онтологии верхнего уровня (Standard Upper Ontology Knowledge Interchange Format)

4.2 Понятия онтологий, определенные в данном стандарте, выделены курсивом. Слитное написание понятий в одно слово применяется при описании классов онтологий и отношений между ними на языке OWL.

5 Онтология SUMO

5.1 Онтология SUMO — это онтология верхнего уровня, которая определяет основные онтологические категории во всех предметных областях (доменах).

5.2 В данном разделе приведен лишь краткий обзор основных концепций SUMO, показанных на рисунке 1.

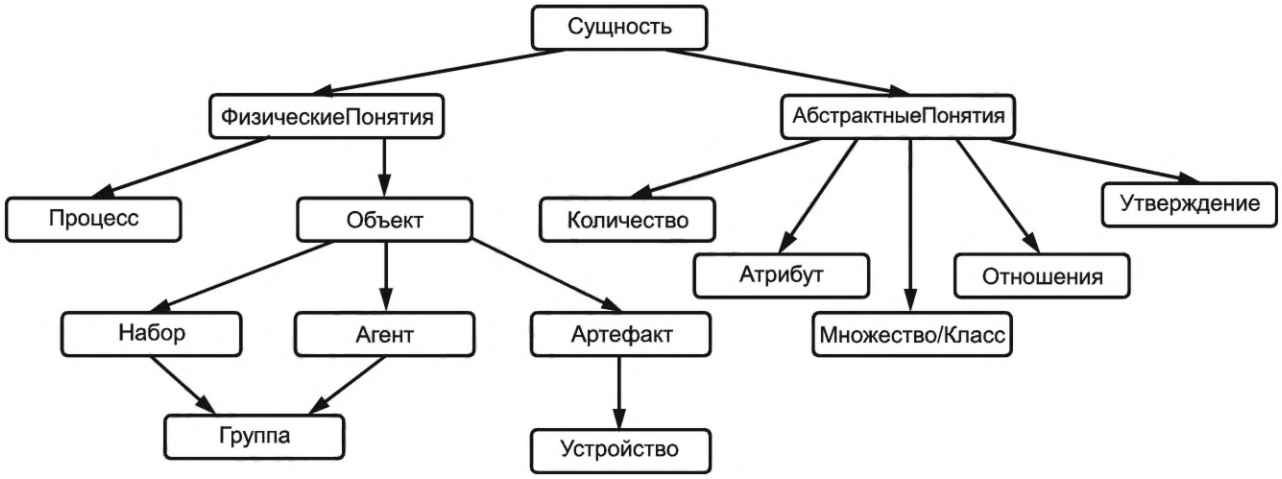


Рисунок 1 — Базовая таксономия SUMO

Базовая таксономия понятий SUMO, записанная в соответствии с требованиями ГОСТ Р ИСО/МЭК 21838-1 на языке OWL:

```
Declaration(Class(:АбстрактныеПонятия))
Declaration(Class(:Агент))
Declaration(Class(:Артефакт))
Declaration(Class(:Атрибут))
Declaration(Class(:Группа))
Declaration(Class(:Количество))
Declaration(Class(:Набор))
Declaration(Class(:Объект))
Declaration(Class(:Отношения))
Declaration(Class(:Процесс))
Declaration(Class(:Сущность))
Declaration(Class(:Устройство))
Declaration(Class(:Утверждение))
Declaration(Class(:ФизическиеПонятия))
##############################
# Classes
##############################
# Class: :АбстрактныеПонятия (:АбстрактныеПонятия)
SubClassOf(:АбстрактныеПонятия :Сущность)
DisjointClasses(:АбстрактныеПонятия :ФизическиеПонятия)
# Class: :Агент (:Агент)
SubClassOf(:Агент :Объект)
# Class: :Артефакт (:Артефакт)
SubClassOf(:Артефакт :Объект)
# Class: :Атрибут (:Атрибут)
SubClassOf(:Атрибут :АбстрактныеПонятия)
# Class: :Группа (:Группа)
SubClassOf(:Группа :Агент)
SubClassOf(:Группа :Набор)
# Class: :Количество (:Количество)
SubClassOf(:Количество :АбстрактныеПонятия)
# Class: :Набор (:Набор)
SubClassOf(:Набор :Объект)
# Class: :Объект (:Объект)
SubClassOf(:Объект :ФизическиеПонятия)
```

```
# Class: :Отношения (:Отношения)
SubClassOf(:Отношения :АбстрактныеПонятия)
# Class: :Процесс (:Процесс)
SubClassOf(:Процесс :ФизическиеПонятия)
# Class: :Устройство (:Устройство)
SubClassOf(:Устройство :Артефакт)
# Class: :Утверждение (:Утверждение)
SubClassOf(:Утверждение :АбстрактныеПонятия)
# Class: :ФизическиеПонятия (:ФизическиеПонятия)
SubClassOf(:ФизическиеПонятия :Сущность)
```

5.3 Основной категорией SUMO является *Сущность* (*Entity*), которая представляет собой непересекающееся разделение *Физических* (*Physical*) и *Абстрактных* (*Abstract*) понятий. *ФизическиеПонятия* представляют сущности, которые имеют пространственно-временное расширение. *АбстрактныеПонятия* описывают сущности, которые не имеют или выбраны так, чтобы не иметь пространственно-временного расширения.

5.4 *ФизическиеПонятия* далее разделяются на *Объект* (*Object*) и *Процесс* (*Process*). Объект существует в пространстве, его отдельные пространственные части существуют во времени параллельно друг другу. *Процесс* — это класс отдельных сущностей, которые происходят во времени и имеют временные (темпоральные) части или стадии.

5.5 *АбстрактныеПонятия* далее подразделяются на *Количество* (*Quantity*), *Атрибут* (*Attribute*), *МножествоИлиКласс* (*SetOrClass*), *Отношения* (*Relation*) и *Утверждение* (*Proposition*). *Количество* абстрагирует числовые и физические величины. *Атрибут* — качества, которые нельзя воплотить в подклассы объекта или было решено не воплощать их в такие подклассы. *МножествоИлиКласс* абстрагирует объекты, которые имеют элементы или экземпляры. *Отношения* абстрагируют *n*-арные зависимости, функции и списки. Наконец, *Утверждение* выражает законченную мысль или набор таких мыслей, которые могут быть представлены такими объектами, как формула или рисунок.

5.6 Введены следующие определения терминов в соответствии с версией SUMO 1.52:

5.6.1 **абстрактные понятия** (abstract): Свойства или качества, отличающиеся от любого конкретного воплощения свойств/качеств в физической среде.

Примечание — Можно сказать, что абстрактные понятия существуют в том же смысле, что и математические объекты, такие как множества и зависимости, но они не могут существовать в определенном месте и времени без какого-либо физического кодирования или воплощения.

5.6.2 **агент** (agent): Что-то или кто-то, способный действовать самостоятельно, и производить изменения в мире.

5.6.3 **артефакт** (artifact): Искусственно созданный трехмерный объект.

5.6.4 **атрибут** (attribute): Качества, которые нельзя выбрать для преобразования в подклассы объекта или было решено не выбирать их для такого преобразования.

5.6.5 **бинарная зависимость** (binary relation): Зависимость, которая верна только для пар вещей, представленная в виде слотов в системах фреймов.

5.6.6 **бинарный предикат** (binary predicate): Предикат, относящийся к двум элементам, его валентность равна двум.

5.6.7 **временной интервал** (time interval): Интервал времени, который имеет как протяженность, так и местоположение на универсальной временной шкале.

5.6.8 **группа** (group): Совокупность агентов.

5.6.9 **измерительное устройство** (measuring device): Любое устройство, предназначенное для измерения физической величины.

5.6.10 **источник питания** (power source): Источник электрической энергии.

5.6.11 **карданное соединение** (universal joint): Шарнир, который соединяет два вала под переменными углами относительно друг друга.

Примечание — Карданное соединение отличается от шарнира равных угловых скоростей тем, что валы не движутся с постоянной скоростью по отношению друг к другу. Работа такого шарнира вызывает вибрацию и износ. Однако карданные соединения изготовить проще, чем шарниры равных угловых скоростей.

5.6.12 **колесо** (wheel): Артефакт, который является деталью механизмов и служит для передачи или преобразования вращательного движения.

5.6.13 **количество** (quantity): Любое указание на количество чего-либо.

Примечание — Существует два подкласса количества: число (в штуках) и физическое количество в единицах измерения (кг, л, м и т. д.).

5.6.14 **компьютерный процесс** (computer process): Процесс обработки информации в компьютере.

5.6.15 **механическое соединение** (mechanical joint): Устройство, которое соединяет две части физической системы и позволяет им перемещаться относительно друг друга.

Примечание — Примерами могут служить петли, направляющие выдвижных ящиков и шаровые шарниры.

5.6.16 **множество или класс** (set or class): Множество или класс множеств и классов; то есть любой экземпляр класса *Абстрактное*, который имеет элементы или экземпляры.

5.6.17 **момент времени** (time point): Точка без расширения на универсальной временной шкале.

Примечание — Моменты времени, в которых происходят процессы, могут быть известны с различной степенью точности и аппроксимации, концептуально подобны точкам, а не интервалам.

5.6.18 **набор** (collection): Коллекции содержащие элементы, подобные *Классам*, но, в отличие от *Классов*, имеющие местоположение в пространстве-времени, при этом, элементы можно добавлять и вычитать, не изменяя таким образом идентичность коллекции.

Примечание — Некоторые примеры: наборы инструментов, футбольные команды.

5.6.19 **область** (region): Топографическое местоположение, охватывающее поверхности объектов, воображаемые места и географические области.

Примечание — Область — это единственный вид объекта, который может включать самого себя.

5.6.20 **объект** (object): Примерно соответствует классу обычных объектов.

Примечание — Примерами являются обычные физические объекты, географические регионы, местоположения выполнения процессов и дополнения объектов в физическом классе.

5.6.21 **однозначное отношение** (single-valued relation): Зависимость является однозначной только в том случае, если присвоение значений каждой позиции аргумента, кроме последней, определяет не более одного значения для последней позиции аргумента.

5.6.22 **отношения** (relation): Класс отношений, включающий три вида отношений: предикат, функция и список.

Примечание — Предикаты и функции обозначают наборы упорядоченных *n*-кортежей. Разница между этими двумя классами заключается в том, что предикаты охватывают операторы формирования формул, в то время как функции охватывают операторы формирования терминов. С другой стороны, список — это определенный упорядоченный *n*-кортеж.

5.6.23 **показатель времени** (time measure): Класс временных длительностей, положений моментов времени и временных интервалов вдоль универсальной шкалы времени.

5.6.24 **процесс** (process): Класс явлений, которые происходят и имеют временные (темпоральные) части или стадии.

Примечание — Примеры включают длительные события, такие как футбольный матч или гонка, такие действия, как преследование и ожидание. Формальное определение таково: все, что происходит во времени, но не является объектом.

5.6.25 **свойство** (property): Этот предикат выполняется между экземпляром *Сущности* и экземпляром *Атрибута*.

5.6.26 **сущность** (entity): Универсальный класс индивидуумов, корневой узел онтологии.

5.6.27 **трехзначный предикат** (ternary predicate): класс предикатов, для которых требуется ровно три аргумента.

5.6.28 **унарная функция** (unary function): Класс функций, для которых требуется один аргумент.

5.6.29 **устройство** (device): *Артефакт*, предназначение которого состоит в том, чтобы служить инструментом в определенном подклассе процесса.

5.6.30 **утверждение** (proposition): Абстрактные сущности, которые выражают законченную мысль или набор таких мыслей.

Примечание — В качестве примера: формула «(экземпляр Робот Федор)» выражает утверждение о том, что сущность с именем Федор является элементом класса Робот.

5.6.31 **физическая величина** (physical quantity): Мера некоторого количественного аспекта моделируемого мира, такого как «диаметр земли» (постоянная длина) и «напряжение в нагруженном деформируемом твердом теле» (мера напряжения, которая является функцией трех пространственных координат).

5.6.32 **физические понятия** (physical): Сущности, имеющие местоположение в пространстве-времени.

Примечание — Следует обратить внимание, что сами местоположения понимаются как имеющие местоположение в пространстве-времени.

5.6.33 **функция** (function): Терминообразующая зависимость, которая отображает *n*-кортеж аргументов в диапазон и связывает этот *n*-кортеж не более чем с одним элементом диапазона.

Примечание — Следует обратить внимание, что диапазон представляет собой множество или класс, и каждый элемент диапазона является экземпляром множества или класса.

5.6.34 **характеристика** (measure): Общий предикат для утверждения, что конкретный объект измеряется определенной физической величиной.

Примечание — В общем случае вторым аргументом этого предиката будет термин, созданный с помощью функции *Измерение*.

5.6.35 **электрическое устройство** (electric device): Устройство, приводимое в действие с помощью электричества.

6 Аксиомы CORAX

6.1 Общие положения

Онтология CORAX определяет понятия, которые являются слишком общими для включения в онтологию CORA, тем самым эффективно расширяя охват множества онтологий, составляющих этот стандарт. Эти концепции охватывают аспекты реальности, которые необходимы для моделирования, но не охватываются явно или полностью онтологией SUMO.

6.2 ТехническийОблик (Design)

Понятие *ТехническийОблик* (*Design*) изделия важно во многих областях. Оно используется для обобщения информации в таких контекстах, как промышленная робототехника.

Технический облик представляет собой *Утверждение* (*Proposition*). Понятия, связанные с *ТехническимОбликом и ОбъектомПроектирования* показаны на рисунке 2.

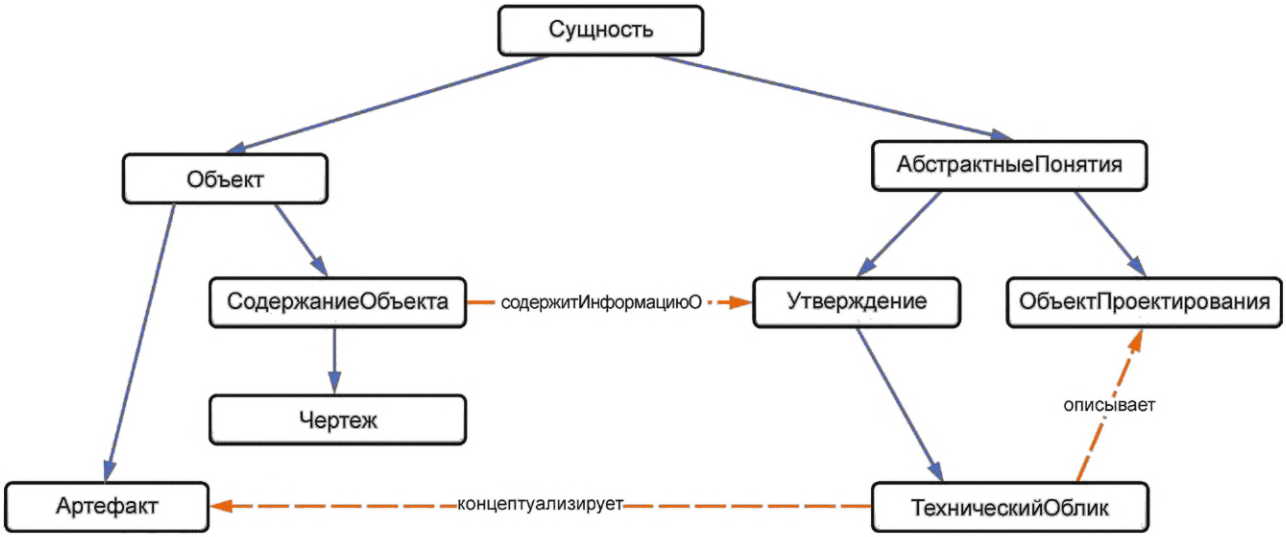


Рисунок 2 — Понятия, связанные с *ТехническимОбликом и ОбъектомПроектирования*

Понятия, связанные с *ТехническимОбликом и ОбъектомПроектирования* на языке OWL:

```
Declaration(Class(:АбстрактныеПонятия))
Declaration(Class(:Артефакт))
Declaration(Class(:ОбъектПроектирования))
Declaration(Class(:Объекты))
Declaration(Class(:СодержаниеОбъекта))
Declaration(Class(:Сущность))
Declaration(Class(:ТехническийОблик))
Declaration(Class(:Утверждение))
Declaration(Class(:Чертеж))
Declaration(ObjectProperty(:концептуализирует))
Declaration(ObjectProperty(:описывает))
Declaration(ObjectProperty(:содержитИнформациюО))
#############################
# Object Properties
#############################
# Object Property: :концептуализирует (:концептуализирует)
AsymmetricObjectProperty(:концептуализирует)
ObjectPropertyDomain(:концептуализирует :ТехническийОблик)
ObjectPropertyRange(:концептуализирует :Артефакт)
# Object Property: :описывает (:описывает)
AsymmetricObjectProperty(:описывает)
ObjectPropertyDomain(:описывает :ТехническийОблик)
ObjectPropertyRange(описывает :ОбъектПроектирования)
# Object Property: :содержитИнформациюО (:содержитИнформациюО)
ObjectPropertyDomain(:содержитИнформациюО :СодержаниеОбъекта)
ObjectPropertyRange(:содержитИнформациюО :Утверждение)
#############################
# Classes
#############################
# Class: :АбстрактныеПонятия (:АбстрактныеПонятия)
SubClassOf(:АбстрактныеПонятия :Сущность)
# Class: :Артефакт (:Артефакт)
SubClassOf(:Артефакт :Объекты)
# Class: :ОбъектПроектирования (:ОбъектПроектирования)
SubClassOf(:ОбъектПроектирования :АбстрактныеПонятия)
```

```
# Class: :Объекты (:Объекты)
SubClassOf(:Объекты :Сущность)
# Class: :СодержаниеОбъекта (:СодержаниеОбъекта)
SubClassOf(:СодержаниеОбъекта :Объекты)
# Class: :ТехническийОблик (:ТехническийОблик)
SubClassOf(:ТехническийОблик :Утверждение)
# Class: :Утверждение (:Утверждение)
SubClassOf(:Утверждение :АбстрактныеПонятия)
# Class: :Чертеж (:Чертеж)
SubClassOf(:Чертеж :СодержаниеОбъекта)
```

Технический облик концептуализирует предполагаемую структуру одного или нескольких артефактов.

Технический облик связан отношением *описывает* с объектами проектирования. Объекты проектирования представляют собой абстрактные идеализации отдельных артефактов, соответствующих техническому облику. Отношение *описывает* декларирует, что утверждение устанавливает факты о некоторой сущности. В CORAX данное отношение показывает, что технический облик устанавливает факты об объекте проектирования.

Объекты проектирования имеют атрибуты и характеристики, которые будут принадлежать создаваемым артефактам. Абстрактные аналоги отношений *Атрибут* (*Attribute*) и *Характеристика* (*Measure*) фиксируют эти понятия.

Объекты проектирования могут состоять из других объектов проектирования, например таких, которые отражают структуру проектируемого артефакта.

Онтология SUMO констатирует предложения, которые могут быть представлены физическими носителями контента, такими как лингвистическое предложение (включая логические предложения), документ, график или компьютерный файл САПР. Этот стандарт не налагает никаких ограничений на типы физических носителей контента, которые могут представлять технический облик.

6.3 ФизическаяСреда (PhysicalEnvironment)

ФизическаяСреда (*PhysicalEnvironment*) — это объект, имеющий по крайней мере одну конкретную часть: область, в которой она расположена.

Кроме того, физическая среда относится, по меньшей мере, к одному эталонному объекту, на основе которого определяется ее область. Понятия, связанные с *ФизическойСредой* показаны на рисунке 3.

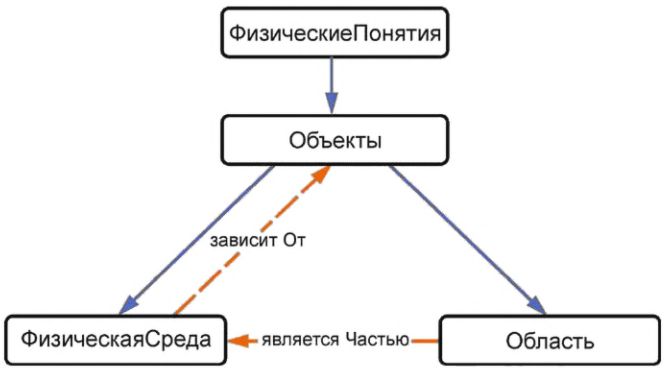


Рисунок 3 — Понятия, связанные с *ФизическойСредой*

Понятия, связанные с *ФизическойСредой* на языке OWL:

```
Declaration(Class(:Область))
Declaration(Class(:Объекты))
Declaration(Class(:ФизическаяСреда))
Declaration(Class(:ФизическиеПонятия))
```

```
Declaration(ObjectProperty(:зависитОт))
Declaration(ObjectProperty(:являетсяЧастью))
############################
# Object Properties
############################
# Object Property: :зависитОт (:зависитОт)
ObjectPropertyDomain(:зависитОт :ФизическаяСреда)
ObjectPropertyRange(:зависитОт :Объекты)
# Object Property: :являетсяЧастью (:ЯвляетсяЧастью)
SubObjectPropertyOf(:являетсяЧастью owl:topObjectProperty)
ObjectPropertyDomain(:являетсяЧастью :Область)
ObjectPropertyRange(:являетсяЧастью :ФизическаяСреда)
############################
# Classes
############################
# Class: :Область (:Область)
SubClassOf(:Область :Объекты)
# Class: :Объекты (:Объекты)
SubClassOf(:Объекты :ФизическиеПонятия)
# Class: :ФизическаяСреда (:ФизическаяСреда)
SubClassOf(:ФизическаяСреда :Объекты)
```

Например, подводный робот работает в *Физической среде*, ограниченной морским дном и поверхностью моря по вертикали, а по горизонтали — только абстрактными измерениями. В этом случае эталонными объектами, которые определяют окружающую среду, являются морское дно, морская поверхность и ориентиры, используемые для определения границы горизонтальной плоскости, в пределах которой будет работать подводный аппарат.

Все объекты, являющиеся частью *Физической среды*, расположены в области, являющейся частью этой среды.

6.4 Взаимодействие (Interaction)

Онтология SUMO не определяет понятие *Взаимодействие* (*Interaction*). Взаимодействие — это *Процесс*, в котором участвуют два агента.

Он состоит из двух подпроцессов, определяющих действие и реакцию. Подпроцесс действия, инициируемый агентом X для реципиента Y, вызывает подпроцесс реакции, имеющий Y в качестве агента и X в качестве реципиента, то есть между ними существует *взаимодействие*.

6.5 ИскусственнаяСистема (ArtificialSystem)

ИскусственнаяСистема (*ArtificialSystem*) — это *Артефакт*, сформированный различными *Устройствами* (и другими *Объектами*), которые *взаимодействуют* для выполнения определенной функции. Для любой части искусственной системы существует, по крайней мере, еще одна часть, с которой она взаимодействует.

Понятия, связанные с *ИскусственнойСистемой* и *Взаимодействием* показаны на рисунке 4.

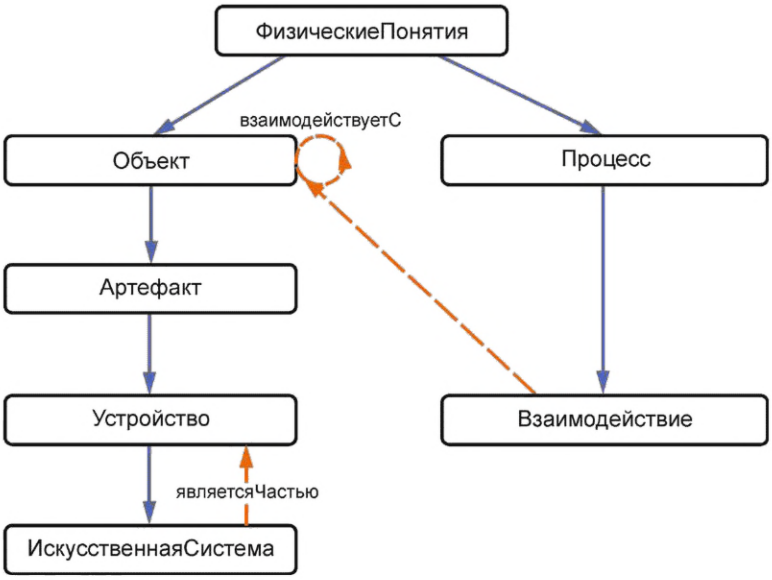


Рисунок 4 — Понятия, связанные с *ИскусственнойСистемой* и *Взаимодействием*

Понятия, связанные с *ИскусственнойСистемой* и *Взаимодействием* на языке OWL:

```
Declaration(Class(:Артефакт))
Declaration(Class(:Взаимодействие))
Declaration(Class(:ИскусственнаяСистема))
Declaration(Class(:Объекты))
Declaration(Class(:Процессы))
Declaration(Class(:Устройство))
Declaration(Class(:ФизическиеПонятия))
Declaration(ObjectProperty(:взаимодействуетС))
Declaration(ObjectProperty(:являетсяЧастью))
##############################
# Object Properties
##############################
# Object Property: :взаимодействуетС (:взаимодействуетС)
SymmetricObjectProperty(:взаимодействуетС)
ObjectPropertyDomain(:взаимодействуетС :Объекты)
ObjectPropertyRange(:взаимодействуетС :Объекты)
# Object Property: :являетсяЧастью (:являетсяЧастью)
ObjectPropertyDomain(:являетсяЧастью :ИскусственнаяСистема)
ObjectPropertyRange(:являетсяЧастью :Устройство)
##############################
# Classes
##############################
# Class: :Артефакт (:Артефакт)
SubClassOf(:Артефакт :Объекты)
# Class: :Взаимодействие (:Взаимодействие)
EquivalentClasses(:Взаимодействие ObjectSomeValuesFrom(:взаимодействуетС
:Объекты))
SubClassOf(:Взаимодействие :Процессы)
# Class: :ИскусственнаяСистема (:ИскусственнаяСистема)
SubClassOf(:ИскусственнаяСистема :Устройство)
# Class: :Объекты (:Объекты)
SubClassOf(:Объекты :ФизическиеПонятия)
```

```
# Class: :Процессы (:Процессы)
SubClassOf(:Процессы :ФизическиеПонятия)
# Class: :Устройство (:Устройство)
SubClassOf(:Устройство :Артефакт)
```

6.6 ОбрабатывающееУстройство (ProcessingDevice)

ОбрабатывающееУстройство (*ProcessingDevice*) — это *ЭлектрическоеУстройство*, предназначенное для использования в качестве *инструмента* в подклассе компьютерных процессов.

Понятия, связанные с *ОбрабатывающимУстройством* на языке OWL:

```
Declaration(Class(:КомпьютерныйПроцесс))
Declaration(Class(:ОбрабатывающееУстройство))
Declaration(Class(:Процесс))
Declaration(Class(:Устройство))
Declaration(Class(:ЭлектрическоеУстройство))
Declaration(ObjectProperty(:являетсяИнструментом))
##############################
# Object Properties
##############################
# Object Property: :являетсяИнструментом (:являетсяИнструментом)
ObjectPropertyDomain(:являетсяИнструментом :ОбрабатывающееУстройство)
ObjectPropertyRange(:являетсяИнструментом :КомпьютерныйПроцесс)
##############################
# Classes
##############################
# Class: :КомпьютерныйПроцесс (:КомпьютерныйПроцесс)
SubClassOf(:КомпьютерныйПроцесс :Процесс)
# Class: :ОбрабатывающееУстройство (:ОбрабатывающееУстройство)
SubClassOf(:ОбрабатывающееУстройство :ЭлектрическоеУстройство)
# Class: :ЭлектрическоеУстройство (:ЭлектрическоеУстройство)
SubClassOf(:ЭлектрическоеУстройство :Устройство)
```

6.7 ДвижениеРобота (RobotMotion)

ДвижениеРобота (*RobotMotion*) — это любой процесс движения, в котором агентом является робот, а реципиентом — компонент робота. Конкретнее, это любой процесс перемещения роботом своих компонентов.

Понятия, связанные с *ДвижениемРобота* на языке OWL:

```
Declaration(Class(:Движение))
Declaration(Class(:ДвижениеРобота))
Declaration(Class(:КомпонентРобота))
Declaration(Class(:Робот))
Declaration(ObjectProperty(:перемещает))
##############################
# Object Properties
##############################
# Object Property: :перемещает (:перемещает)
ObjectPropertyDomain(:перемещает :Робот)
ObjectPropertyRange(:перемещает :КомпонентРобота)
##############################
# Classes
##############################
# Class: :ДвижениеРобота (:ДвижениеРобота)
EquivalentClasses(:ДвижениеРобота ObjectSomeValuesFrom(:Перемещает
:КомпонентРобота))
SubClassOf(:ДвижениеРобота :Движение)
# Class: :КомпонентРобота (:КомпонентРобота)
SubClassOf(:КомпонентРобота :Робот)
```

6.8 КоммуникацияЧеловекРобот (HumanRobotCommunication)

КоммуникацияЧеловекРобот (*HumanRobotCommunication*) — это любой процесс, связанный с передачей информации между людьми и роботами.

6.9 КоммуникацияРоботРобот (RobotRobotCommunication)

КоммуникацияРоботРобот (*RobotRobotCommunication*) — это любой процесс, связанный с передачей информации между двумя или более роботами.

Таксономия понятий *КоммуникацияЧеловекРобот и КоммуникацияРоботРобот* на языке OWL:

```
Declaration(Class(:Агенты))
Declaration(Class(:Взаимодействие))
Declaration(Class(:КоммуникацияРоботРобот))
Declaration(Class(:КоммуникацияЧеловекРобот))
Declaration(Class(:Объекты))
Declaration(Class(:Процессы))
Declaration(Class(:Робот))
Declaration(Class(:Человек))
Declaration(ObjectProperty(:коммуникацияРоботРобот))
Declaration(ObjectProperty(:коммуникацияЧеловекРобот))
Declaration(ObjectProperty(:взаимодействуетС))
##############################
# Object Properties
##############################
# Object Property: :коммуникацияРоботРобот (:коммуникацияРоботРобот)
SubObjectPropertyOf(:коммуникацияРоботРобот :взаимодействуетС)
SymmetricObjectProperty(:коммуникацияРоботРобот)
IrreflexiveObjectProperty(:коммуникацияРоботРобот)
ObjectPropertyDomain(:коммуникацияРоботРобот :Робот)
ObjectPropertyRange(:коммуникацияРоботРобот :Робот)
# Object Property: :коммуникацияЧеловекРобот (:коммуникацияЧеловекРобот)
SubObjectPropertyOf(:коммуникацияЧеловекРобот :взаимодействуетС)
SymmetricObjectProperty(:коммуникацияЧеловекРобот)
ObjectPropertyDomain(:коммуникацияЧеловекРобот :Человек)
ObjectPropertyRange(:коммуникацияЧеловекРобот :Робот)
# Object Property: :взаимодействуетС (:взаимодействуетС)
SymmetricObjectProperty(:взаимодействуетС)
##############################
# Classes
##############################
# Class: :Агенты (:Агенты)
SubClassOf(:Агенты :Объекты)
# Class: :Взаимодействие (:Взаимодействие)
SubClassOf(:Взаимодействие :Процессы)
# Class: :КоммуникацияРоботРобот (:КоммуникацияРоботРобот)
EquivalentClasses(:КоммуникацияРоботРобот ObjectSomeValuesFrom(:КоммуникацияРобо
тРобот :Робот))
SubClassOf(:КоммуникацияРоботРобот :Взаимодействие)
DisjointClasses(:КоммуникацияРоботРобот :КоммуникацияЧеловекРобот)
# Class: :КоммуникацияЧеловекРобот (:КоммуникацияЧеловекРобот)
EquivalentClasses(:КоммуникацияЧеловекРобот ObjectSomeValuesFrom(:КоммуникацияЧе
ловекРобот :Робот))
EquivalentClasses(:КоммуникацияЧеловекРобот ObjectSomeValuesFrom(:КоммуникацияЧе
ловекРобот :Человек))
SubClassOf(:КоммуникацияЧеловекРобот :Взаимодействие)
# Class: :Робот (:Робот)
SubClassOf(:Робот :Агенты)
```

```
# Class: :Человек (:Человек)
SubClassOf(:Человек :Агенты)
```

7 Аксиомы CORA

7.1 Робот (Robot)

Робот (*Robot*) является основным понятием в CORA. Это понятие связано с большинством других понятий в этой онтологии.

Таксономия основных понятий в CORA и их связь с другими онтологиями показана на рисунке 5.

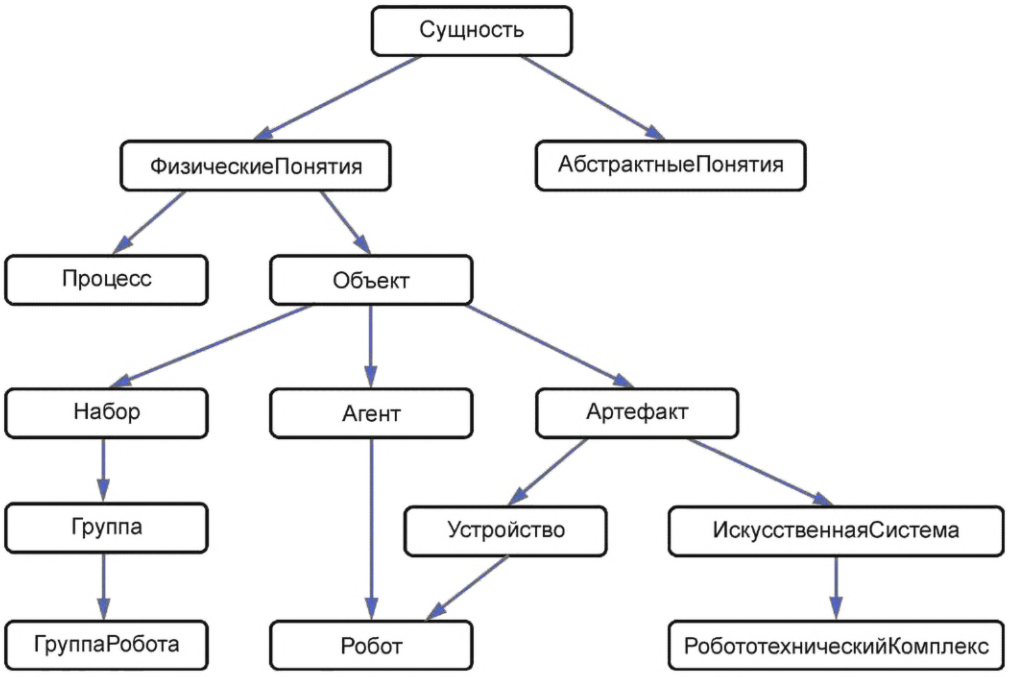


Рисунок 5 — Таксономия основных понятий в CORA и их связь с другими онтологиями

Во-первых, *Робот* — это *Устройство*. Онтология SUMO определяет Устройство как артефакт, предназначенный для использования в качестве инструмента в подклассе процесса. В некоторых случаях, например при использовании многоцелевых роботов, класс процесса может быть неизвестен априори.

Робот также является *Агентом*. Роботы выполняют задачи путем воздействия на окружающую среду или на самих себя.

Таксономия основных понятий в CORA на языке OWL:

```
Declaration(Class(:АбстрактныеПонятия))
Declaration(Class(:Агент))
Declaration(Class(:Артефакт))
Declaration(Class(:Группа))
Declaration(Class(:ГруппаРоботов))
Declaration(Class(:ИскусственнаяСистема))
Declaration(Class(:Набор))
Declaration(Class(:Объекты))
Declaration(Class(:Процессы))
Declaration(Class(:Робот))
Declaration(Class(:РобототехническийКомплекс))
Declaration(Class(:Сущность))
```

```
Declaration(Class(:Устройство))
Declaration(Class(:ФизическиеПонятия))
############################
# Classes
############################
# Class: :АбстрактныеПонятия (:АбстрактныеПонятия)
SubClassOf(:АбстрактныеПонятия :Сущность)
# Class: :Агент (:Агент)
SubClassOf(:Агент :Объекты)
# Class: :Артефакт (:Артефакт)
SubClassOf(:Артефакт :Объекты)
# Class: :Группа (:Группа)
SubClassOf(:Группа :Набор)
# Class: :ГруппаРоботов (:ГруппаРоботов)
SubClassOf(:ГруппаРоботов :Группа)
# Class: :ИскусственнаяСистема (:ИскусственнаяСистема)
SubClassOf(:ИскусственнаяСистема :Артефакт)
# Class: :Набор (:Набор)
SubClassOf(:Набор :Объекты)
# Class: :Объекты (:Объекты)
SubClassOf(:Объекты :ФизическиеПонятия)
# Class: :Процессы (:Процессы)
SubClassOf(:Процессы :ФизическиеПонятия)
# Class: :Робот (:Робот)
SubClassOf(:Робот :Агент)
SubClassOf(:Робот :Устройство)
# Class: :РобототехническийКомплекс (:РобототехническийКомплекс)
SubClassOf(:РобототехническийКомплекс :ИскусственнаяСистема)
# Class: :Устройство (:Устройство)
SubClassOf(:Устройство :Артефакт)
# Class: :ФизическиеПонятия (:ФизическиеПонятия)
SubClassOf(:ФизическиеПонятия :Сущность)
```

7.2 КомпонентРобота (robotPart)

Роботы содержат в себе другие устройства в качестве частей самих себя. Существует множество устройств, которые могут выполнять функции *КомпонентаРобота* (*robotPart*), и невозможно заранее определить, какие устройства могут быть компонентами робота, а какие не могут.

Устройства, которые считаются компонентами робота, когда они установлены на роботе, не всегда являются компонентом робота в онтологическом смысле, поскольку они существуют сами по себе и в большинстве случаев могут быть связаны с другими типами устройств. Например, источник питания — это, по сути, устройство. Однако конкретный экземпляр источника питания может динамически рассматриваться как *КомпонентРобота* в течение определенного интервала времени, когда этот источник подключен к роботу. Это представлено через отношение часть, которая связывает устройства и роботы.

7.3 ИнтерфейсРобота (RobotInterface)

Робот взаимодействует с окружающим его миром через интерфейс.

ИнтерфейсРобота (*RobotInterface*) — это устройство, состоящее из сенсорных, исполнительных и коммуникационных компонентов. Через интерфейс робот может воспринимать окружающую среду и воздействовать на нее, а также коммуницировать с другими агентами. Таким образом, интерфейс робота можно рассматривать как способ обозначения всех устройств, которые позволяют роботу взаимодействовать с окружающим миром.

Каждый сенсорный компонент данного робота является частью его интерфейса. Каждый исполнительный компонент данного робота является частью его интерфейса. Каждый коммуникационный компонент данного робота является частью его интерфейса.

Каждый интерфейс робота должен иметь компонент, который является либо сенсорным компонентом робота, либо исполнительным компонентом робота, либо коммуникационным компонентом робота.

Каждый робот имеет один и только один интерфейс.

Таксономия понятия *ИнтерфейсРобота* на языке OWL:

```
Declaration(Class(:ИнтерфейсРобота))
Declaration(Class(:ИсполнительныйКомпонентРобота))
Declaration(Class(:КоммуникационныйКомпонентРобота))
Declaration(Class(:КомпонентыРобота))
Declaration(Class(:Робот))
Declaration(Class(:СенсорныйКомпонентРобота))
Declaration(Class(:Устройства))
Declaration(ObjectProperty(:выполняетФункцию))
Declaration(ObjectProperty(:имеет))
Declaration(ObjectProperty(:являетсяКомпонентом))
#############################
# Object Properties
#############################
# Object Property: :выполняетФункцию (:выполняетФункцию)
AsymmetricObjectProperty(:выполняетФункцию)
ObjectPropertyDomain(:выполняетФункцию :Устройства)
ObjectPropertyRange(:выполняетФункцию :ИсполнительныйКомпонентРобота)
ObjectPropertyRange(:выполняетФункцию :КоммуникационныйКомпонентРобота)
ObjectPropertyRange(:выполняетФункцию :СенсорныйКомпонентРобота)
# Object Property: :имеет (:имеет)
FunctionalObjectProperty(:имеет)
ObjectPropertyDomain(:имеет :Робот)
ObjectPropertyRange(:имеет :ИнтерфейсРобота)
# Object Property: :являетсяКомпонентом (:являетсяКомпонентом)
AsymmetricObjectProperty(:являетсяКомпонентом)
ObjectPropertyDomain(:являетсяКомпонентом :ИсполнительныйКомпонентРобота)
ObjectPropertyDomain(:являетсяКомпонентом :КоммуникационныйКомпонентРобота)
ObjectPropertyDomain(:являетсяКомпонентом :СенсорныйКомпонентРобота)
ObjectPropertyRange(:являетсяКомпонентом :ИнтерфейсРобота)
#############################
# Classes
#############################
# Class: :ИнтерфейсРобота (:ИнтерфейсРобота)
SubClassOf(:ИнтерфейсРобота :Устройства)
# Class: :ИсполнительныйКомпонентРобота (:ИсполнительныйКомпонентРобота)
SubClassOf(:ИсполнительныйКомпонентРобота :КомпонентыРобота)
# Class: :КоммуникационныйКомпонентРобота (:КоммуникационныйКомпонентРобота)
SubClassOf(:КоммуникационныйКомпонентРобота :КомпонентыРобота)
# Class: :КомпонентыРобота (:КомпонентыРобота)
SubClassOf(:КомпонентыРобота :Робот)
# Class: :СенсорныйКомпонентРобота (:СенсорныйКомпонентРобота)
SubClassOf(:СенсорныйКомпонентРобота :КомпонентыРобота)
```

7.4 Понятия, связанные с автономностью робота

Понятия *ПолностьюАвтономныйРобот* (*fullAutonomousRobot*), *ПолуавтономныйРобот* (*semiAutonomousRobot*), *ТелеуправляемыйРобот* (*teleoperatedRobot*), *Дистанционно-управляемыйРобот* (*remoteControlledRobot*) и *АвтоматическийРобот* (*automatedRobot*) в онтологии CORA определены исходя из режимов работы.

Онтология CORA предполагает, что автономия относится к конкретному процессу. Робот, участвующий в процессе в качестве агента, может участвовать как полностью автономный робот, полуавто-

номный робот, телеуправляемый робот, дистанционно-управляемый робот или автоматический робот. CORA определяет эти различные ситуации как роли, которые робот может взять на себя в процессе.

ПолностьюАвтономныйРобот, *ПолуавтономныйРобот*, *ТелеуправляемыйРобот* и *Дистанционно-управляемыйРобот* представляют роботов, участвующих в качестве агентов в процессах, соответствующих режимам работы, определенным в ALFUS как *полностью автономный, полуавтономный, телеуправляемый* и *дистанционно-управляемый*, соответственно. CORA дополняет этот список понятием *АвтоматическийРобот*, приписываемой роботам, действующим в процессе как автоматы, подобно часовому механизму, неспособному изменить свои действия.

Один и тот же робот может иметь несколько ролей в разных процессах. Например, робот-вездеход, исследующий планету, может выполнять роль полуавтономного робота в процессе исследования планеты, но в то же время он может быть полностью автономным в процессе навигации.

Таксономия понятий, связанных с автономностью робота на языке OWL:

```
############################
# Classes
############################
# Class: :АвтоматическийРобот (:АвтоматическийРобот)
SubClassOf(:АвтоматическийРобот :Агент)
# Class: :Агент (:Агент)
SubClassOf(:Агент :Робот)
# Class: :Дистанционно-управляемыйРобот (:Дистанционно-управляемыйРобот)
SubClassOf(:Дистанционно-управляемыйРобот :Агент)
# Class: :ПолностьюАвтономныйРобот (:ПолностьюАвтономныйРобот)
SubClassOf(:ПолностьюАвтономныйРобот :Агент)
# Class: :ПолуавтономныйРобот (:ПолуавтономныйРобот)
SubClassOf(:ПолуавтономныйРобот :Агент)
# Class: :ТелеуправляемыйРобот (:ТелеуправляемыйРобот)
SubClassOf(:ТелеуправляемыйРобот :Агент)
```

7.5 ГруппаРоботов (RobotGroup)

ГруппаРоботов (*RobotGroup*) — это совокупность агентов, состоящая исключительно из роботов. Агенты, из которых состоит группа, образуют агентское сообщество этой группы.

Примеры групп роботов: футбольные команды роботов или бригады роботов-паяльщиков на фабриках. Часть онтологии, показывающая основные концепции, связанные с *ГруппойРоботов* представлена на рисунке 6.

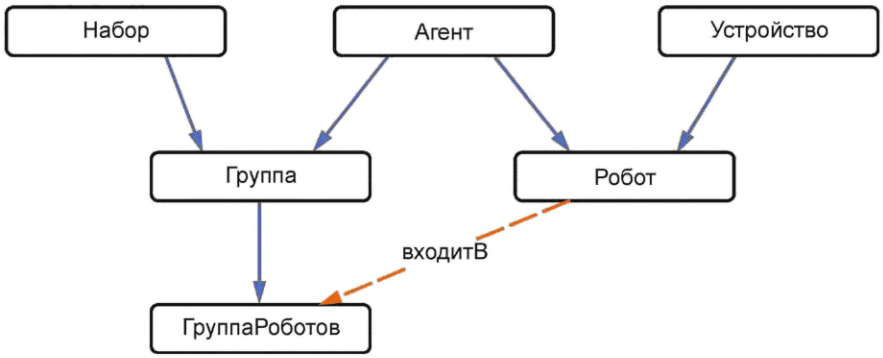


Рисунок 6 — Онтологии, показывающие основные концепции, связанные с *ГруппойРоботов*

Онтологии, связанные с *ГруппойРоботов* на языке OWL:

```
Declaration(Class(:Агент))
Declaration(Class(:Группа))
Declaration(Class(:ГруппаРоботов))
Declaration(Class(:Набор))
Declaration(Class(:Робот))
```

```
Declaration(Class(:Устройство))
Declaration(ObjectProperty(:входитВ))
#############################
# Object Properties
#############################
# Object Property: :входитВ (:входитВ)
ObjectPropertyDomain(:входитВ :Робот)
ObjectPropertyRange(:входитВ :ГруппаРоботов)
#############################
# Classes
#############################
# Class: :Группа (:Группа)
SubClassOf(:Группа :Агент)
SubClassOf(:Группа :Набор)
# Class: :ГруппаРоботов (:ГруппаРоботов)
SubClassOf(:ГруппаРоботов :Группа)
# Class: :Робот (:Робот)
SubClassOf(:Робот :Агент)
SubClassOf(:Робот :Устройство)
```

Термин *ГруппаРоботов* также включает в себя *сложные роботы* (*complex robots*). Они представляют собой механизмы, образованные множеством связанных друг с другом агентов. Например, робот-танк, в котором корпус и башня являются независимыми автономными роботами, которые могут координировать свои действия для достижения общей цели. Робот-змея, состоящая из более мелких автономных роботов, также является примером сложного робота.

7.6 РобототехническийКомплекс (RoboticSystem)

Роботы и другие устройства могут образовывать *РобототехническийКомплекс* (*RoboticSystem*).

РобототехническийКомплекс — это искусственная система, образованная роботами и устройствами, предназначенными для обеспечения выполнения роботом функционального назначения (задания). Взаимоотношения *РобототехническогоКомплекса* с роботом и внешней средой робота показаны на рисунке 7.

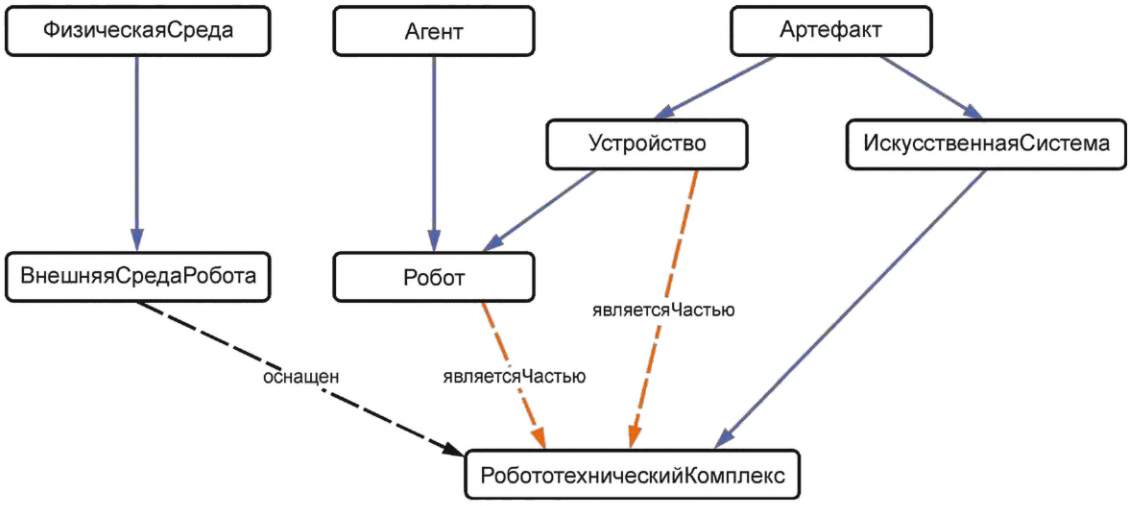


Рисунок 7 — Взаимоотношения *РобототехническогоКомплекса* с роботом и внешней средой робота

Взаимоотношения *РобототехническогоКомплекса* с роботом и внешней средой робота на языке OWL:

```
Declaration(Class(:Агент))
Declaration(Class(:Артефакт))
```

```
Declaration(Class(:ИскусственнаяСистема))
Declaration(Class(:Робот))
Declaration(Class(:РобототехническийКомплекс))
Declaration(Class(:ВнешняяСредаРобота))
Declaration(Class(:Устройство))
Declaration(Class(:ФизическаяСреда))
Declaration(ObjectProperty(:оснащен))
Declaration(ObjectProperty(:являетсяЧастью))
#############################
# Object Properties
#############################
# Object Property: :оснащен (:оснащен)
SubObjectPropertyOf(:оснащен owl:topObjectProperty)
AsymmetricObjectProperty(:оснащен)
ObjectPropertyDomain(:оснащен :ВнешняяСредаРобота)
ObjectPropertyRange(:оснащен :РобототехническийКомплекс)
# Object Property: :являетсяЧастью (:являетсяЧастью)
ObjectPropertyDomain(:являетсяЧастью :Робот)
ObjectPropertyDomain(:являетсяЧастью :Устройство)
ObjectPropertyRange(:являетсяЧастью :РобототехническийКомплекс)
#############################
# Classes
#############################
# Class: :ИскусственнаяСистема (:ИскусственнаяСистема)
SubClassOf(:ИскусственнаяСистема :Артефакт)
# Class: :Робот (:Робот)
SubClassOf(:Робот :Агент)
SubClassOf(:Робот :Устройство)
# Class: :РобототехническийКомплекс (:РобототехническийКомплекс)
SubClassOf(:РобототехническийКомплекс :ИскусственнаяСистема)
# Class: :ВнешняяСредаРобота (:ВнешняяСредаРобота)
SubClassOf(:ВнешняяСредаРобота :ФизическаяСреда)
# Class: :Устройство (:Устройство)
SubClassOf(:Устройство :Артефакт)
```

7.7 ВнешняяСредаРобота (RoboticEnvironment)

Физическая среда, в которой присутствуют робототехнические комплексы, является *ВнешнейСредойРобота* (*RoboticEnvironment*).

Как минимум, одна часть робототехнического комплекса, который присутствует во *ВнешнейСредеРобота*, должна быть расположена в этой среде в некой области.

7.8 ОдинарныйРобототехническийКомплекс (SingleRoboticSystem) и КоллективныйРобототехническийКомплекс (CollectiveRoboticSystem)

Робототехнические комплексы могут содержать одного или несколько роботов. Робототехнические комплексы подразделяются на *ОдинарныеРобототехническиеКомплексы* (*SingleRoboticSystems*) и *КоллективныеРобототехническиеКомплексы* (*CollectiveRoboticSystems*), как показано на рисунке 8.

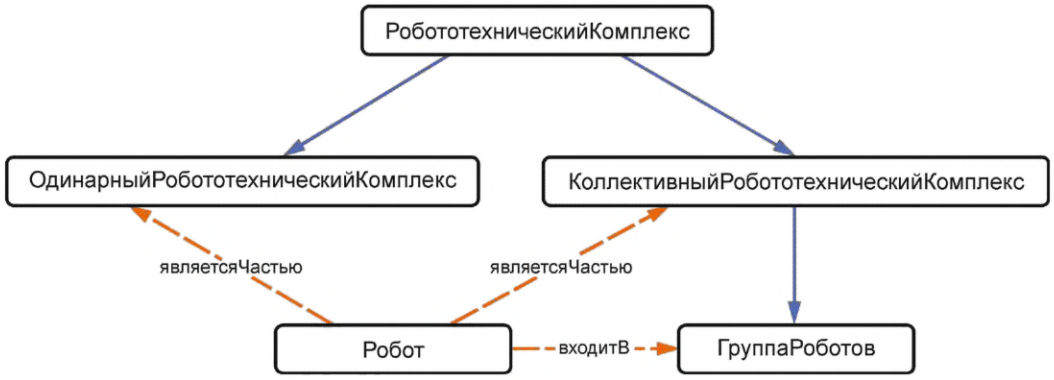


Рисунок 8 — Различные типы робототехнических комплексов

Типы робототехнических комплексов на языке OWL:

```
Declaration(Class(:ГруппаРоботов))
Declaration(Class(:КоллективныйРобототехническийКомплекс))
Declaration(Class(:ОдинарныйРобототехническийКомплекс))
Declaration(Class(:Робот))
Declaration(Class(:РобототехническийКомплекс))
Declaration(ObjectProperty(:входитВ))
Declaration(ObjectProperty(:являетсяЧастью))
#############################
# Object Properties
#############################
# Object Property: :входитВ (:входитВ)
ObjectPropertyDomain(:входитВ :Робот)
ObjectPropertyRange(:входитВ :ГруппаРоботов)
# Object Property: :являетсяЧастью (:являетсяЧастью)
ObjectPropertyDomain(:являетсяЧастью :Робот)
ObjectPropertyRange(:являетсяЧастью :КоллективныйРобототехническийКомплекс)
ObjectPropertyRange(:являетсяЧастью :ОдинарныйРобототехническийКомплекс)
#############################
# Classes
#############################
# Class: :ГруппаРоботов (:ГруппаРоботов)
SubClassOf(:ГруппаРоботов :КоллективныйРобототехническийКомплекс)
# Class: :КоллективныйРобототехническийКомплекс
(:КоллективныйРобототехническийКомплекс)
SubClassOf(:КоллективныйРобототехническийКомплекс :РобототехническийКомплекс)
# Class: :ОдинарныйРобототехническийКомплекс
(:ОдинарныйРобототехническийКомплекс)
SubClassOf(:ОдинарныйРобототехническийКомплекс :РобототехническийКомплекс)
# Class: :Робот (:Робот)
SubClassOf(:Робот :ГруппаРоботов)
SubClassOf(:Робот :КоллективныйРобототехническийКомплекс)
SubClassOf(:Робот :ОдинарныйРобототехническийКомплекс)
```

В *ОдинарномРобототехническомКомплексе* имеется один и только один робот. *КоллективныйРобототехническийКомплекс* состоит из двух или более роботов.

Предполагается, что все роботы в *КоллективномРобототехническомКомплексе* каким-то образом сотрудничают ради достижения общей цели. Естественно, у каждого робота может быть своя локальная цель, но эти цели должны быть подчиненны групповой цели. Поэтому все роботы *КоллективногоРобототехническогоКомплекса* должны быть членами одной группы. Это справедливо даже для тех ситуаций, когда между роботами нет прямого взаимодействия, например на автоматизированной

сборочной линии. В этом случае группа роботов также является частью коллективного робототехнического комплекса.

Другим примером робототехнического комплекса является автоматизированная система домашнего помощника, состоящая из робота-помощника, автономных датчиков и приводов для открытия дверей. В частности, понятие робототехнического комплекса необходимо для описания параметров промышленной роботизации в других онтологиях робототехники.

8 Аксиомы RPARTS

8.1 Общие положения

Онтология CORA утверждает, что любое устройство может быть компонентом робота. В этом смысле устройство является компонентом робота, когда оно присоединено к роботу и выполняет некоторую функцию. Онтология RPARTS объединяет некоторые из наиболее общих и типичных конкретных типов компонетнов робота.

Онтология RPARTS описывает конкретные типы функций, которые относятся к общей функции компонентов робота. Они определяют требования, которые позволяют отнести устройство к тому или иному компоненту робота. Этот набор функций может быть расширен. Данная онтология не описывает реальных устройств, которые могут быть компонентами робота.

8.2 СенсорныйКомпонентРобота (robotSensingPart)

Измерительное устройство, подключенное к роботу, является *СенсорнымКомпонентомРобота* (*robotSensingPart*).

Например, лазерный датчик выполняет функцию сенсорного компонента робота, когда он подключен к роботу.

8.3 ИсполнительныйКомпонентРобота (robotActuatingPart)

Устройство, с помощью которого робот может двигаться и действовать в окружающей среде, является *ИсполнительнымКомпонентомРобота* (*robotActuatingPart*).

Например, захватное устройство выполняет функцию исполнительного компонента робота, когда оно установлено на роботе.

8.4 КоммуникационныйКомпонентРобота (robotCommunicatingPart)

КоммуникационныйКомпонентРобота (*robotCommunicatingPart*) — любое устройство, служащее инструментом в процессе взаимодействия робота с роботом или человека с роботом, позволяя роботу отправлять информацию человеку или другому роботу, или получать информацию от него.

Например, радиоприемник выполняет функцию коммуникационного компонента робота, когда он подключен к роботу.

8.5 ОбрабатывающийКомпонентРобота (robotProcessingPart)

ОбрабатывающийКомпонентРобота (*robotProcessingPart*) — любое вычислительное устройство, с помощью которого робот может обрабатывать информацию.

Например, контроллер выполняет функцию вычислительного компонента робота, когда он подключен к роботу.

Понятия, связанные с компонентами робота на языке OWL:

```
Declaration(Class(:Робот))
Declaration(Class(:КомпонентРобота))
Declaration(Class(:ИсполнительныйКомпонентРобота))
Declaration(Class(:КоммуникационныйКомпонентРобота))
Declaration(Class(:ОбрабатывающийКомпонентРобота))
Declaration(Class(:СенсорныйКомпонентРобота))
Declaration(Class(:Устройства))
Declaration(ObjectProperty(:выполняетФункцию))
############################
```

```
# Object Properties
#############################
# Object Property: :выполняетФункцию (:выполняетФункцию)
ObjectPropertyDomain(:выполняетФункцию :Устройства)
ObjectPropertyRange(:выполняетФункцию :ИсполнительныйКомпонентРобота)
ObjectPropertyRange(:выполняетФункцию :КоммуникационныйКомпонентРобота)
ObjectPropertyRange(:выполняетФункцию :ОбрабатывающийКомпонентРобота)
ObjectPropertyRange(:выполняетФункцию :СенсорныйКомпонентРобота)
#############################
# Classes
#############################
# Class: :КомпонентРобота (:КомпонентРобота)
SubClassOf(:КомпонентРобота :Робот)
# Class: :ИсполнительныйКомпонентРобота (:ИсполнительныйКомпонентРобота)
SubClassOf(:ИсполнительныйКомпонентРобота :КомпонентРобота)
# Class: :КоммуникационныйКомпонентРобота (:КоммуникационныйКомпонентРобота)
SubClassOf(:КоммуникационныйКомпонентРобота :КомпонентРобота)
# Class: :ОбрабатывающийКомпонентРобота (:ОбрабатывающийКомпонентРобота)
SubClassOf(:ОбрабатывающийКомпонентРобота :КомпонентРобота)
# Class: :СенсорныйКомпонентРобота (:СенсорныйКомпонентРобота)
SubClassOf(:СенсорныйКомпонентРобота :КомпонентРобота)
```

9 Аксиомы POS

9.1 Общие положения

POS представляет собой онтологию позиции, ориентации и пространственного расположения. На рисунке 9 показана общая структура концептуальных понятий, представляющих позицию.

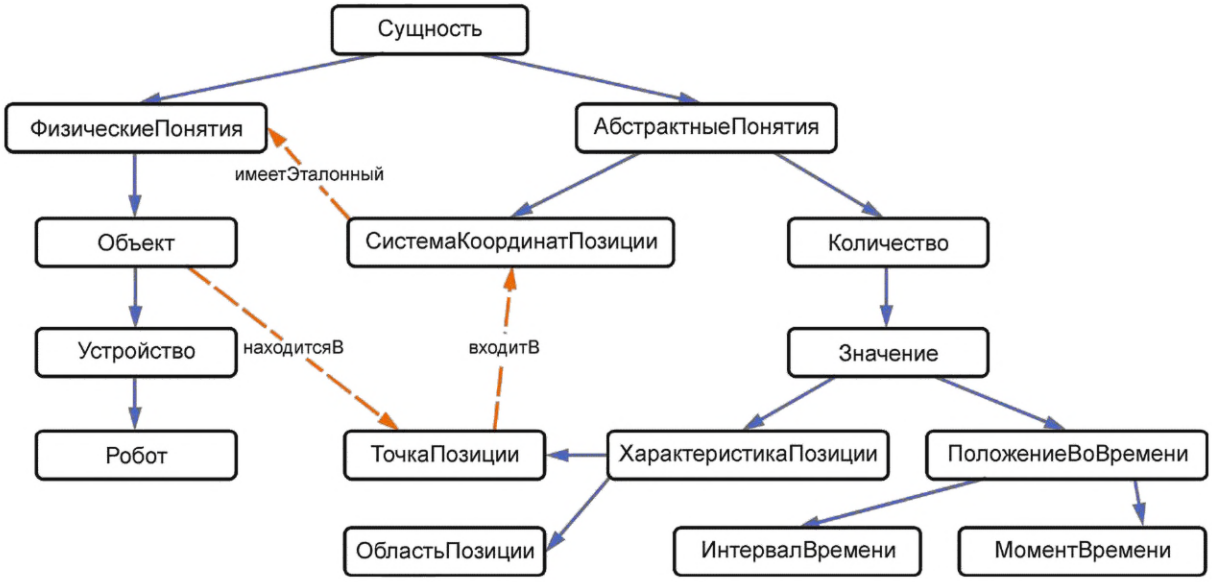


Рисунок 9 — Основные концептуальные понятия онтологии POS

9.2 СистемаКоординатПозиции (PositionCoordinateSystem)

СистемаКоординатПозиции (PositionCoordinateSystem) — это абстрактная сущность, которая используется для указания местоположения по отношению к эталонному объекту.

```
# Class: :СистемаКоординатПозиции (:СистемаКоординатПозиции)
SubClassOf(:СистемаКоординатПозиции :АбстрактныеПонятия)
```

В системе координат позиции координат имеется один эталонный объект.

```
# Object Property: :имеетЭталонный (:имеетЭталонный)
FunctionalObjectProperty(:имеетЭталонный)
AsymmetricObjectProperty(:имеетЭталонный)
ObjectPropertyDomain(:имеетЭталонный :СистемаКоординат)
ObjectPropertyRange(:имеетЭталонный :Объект)
```

Основные концептуальные понятия онтологии POS на языке OWL:

```
Declaration(Class(:АбстрактныеПонятия))
Declaration(Class(:Значение))
Declaration(Class(:ИнтервалВремени))
Declaration(Class(:Количество))
Declaration(Class(:МоментВремени))
Declaration(Class(:ОбластьПозиции))
Declaration(Class(:Объект))
Declaration(Class(:ПоложениеВоВремени))
Declaration(Class(:Робот))
Declaration(Class(:СистемаКоординатПозиции))
Declaration(Class(:Сущность))
Declaration(Class(:ТочкаПозиции))
Declaration(Class(:Устройство))
Declaration(Class(:ФизическиеПонятия))
Declaration(Class(:ХарактеристикаПозиции))
Declaration(ObjectProperty(:входитВ))
Declaration(ObjectProperty(:имеетЭталонный))
Declaration(ObjectProperty(:находитсяВ))
Declaration(ObjectProperty(:относительноеПоложение))
Declaration(NamedIndividual(:ПоложениеЭталонногоОбъекта))
Declaration(NamedIndividual(:ЭталонныйОбъект))
#############################
# Object Properties
#############################
# Object Property: :входитВ (:входитВ)
ObjectPropertyDomain(:входитВ :ТочкаПозиции)
ObjectPropertyRange(:входитВ :СистемаКоординатПозиции)
# Object Property: :имеетЭталонный (:имеетЭталонный)
FunctionalObjectProperty(:имеетЭталонный)
AsymmetricObjectProperty(:имеетЭталонный)
ObjectPropertyDomain(:имеетЭталонный :СистемаКоординатПозиции)
ObjectPropertyRange(:имеетЭталонный :Объект)
# Object Property: :находитсяВ (:находитсяВ)
FunctionalObjectProperty(:находитсяВ)
ObjectPropertyDomain(:находитсяВ :Объект)
ObjectPropertyRange(:находитсяВ :ТочкаПозиции)
# Object Property: :относительноеПоложение (:относительноеПоложение)
ObjectPropertyDomain(:относительноеПоложение ObjectSomeValuesFrom(:находитсяВ
:ХарактеристикаПозиции))
ObjectPropertyRange(:относительноеПоложение ObjectSomeValuesFrom(:находитсяВ
:ХарактеристикаПозиции))
#############################
# Classes
#############################
# Class: :АбстрактныеПонятия (:АбстрактныеПонятия)
SubClassOf(:АбстрактныеПонятия :Сущность)
```

```
# Class: :Значение (:Значение)
SubClassOf(:Значение :Количество)
# Class: :ИнтервалВремени (:ИнтервалВремени)
SubClassOf(:ИнтервалВремени :ПоложениеВоВремени)
# Class: :Количество (:Количество)
SubClassOf(:Количество :АбстрактныеПонятия)
# Class: :МоментВремени (:МоментВремени)
SubClassOf(:МоментВремени :ПоложениеВоВремени)
# Class: :ОбластьПозиции (:ОбластьПозиции)
SubClassOf(:ОбластьПозиции :ХарактеристикаПозиции)
# Class: :Объект (:Объект)
SubClassOf(:Объект :ФизическиеПонятия)
# Class: :ПоложениеВоВремени (:ПоложениеВоВремени)
SubClassOf(:ПоложениеВоВремени :Значение)
# Class: :Робот (:Робот)
SubClassOf(:Робот :Устройство)
# Class: :СистемаКоординатПозиции (:СистемаКоординатПозиции)
SubClassOf(:СистемаКоординатПозиции :АбстрактныеПонятия)
# Class: :ТочкаПозиции (:ТочкаПозиции)
SubClassOf(:ТочкаПозиции :ХарактеристикаПозиции)
# Class: :Устройство (:Устройство)
SubClassOf(:Устройство :Объект)
# Class: :ФизическиеПонятия (:ФизическиеПонятия)
SubClassOf(:ФизическиеПонятия :Сущность)
# Class: :ХарактеристикаПозиции (:ХарактеристикаПозиции)
SubClassOf(:ХарактеристикаПозиции :Значение)
#############################
# Named Individuals
#############################
# Individual: :ПоложениеЭталонногоОбъекта (:ПоложениеЭталонногоОбъекта)
ClassAssertion(:ТочкаПозиции :ПоложениеЭталонногоОбъекта)
# Individual: :ЭталонныйОбъект (:ЭталонныйОбъект)
ClassAssertion(:Объект :ЭталонныйОбъект)
ObjectPropertyAssertion(:находитсяВ :ЭталонныйОбъект
:ПоложениеЭталонногоОбъекта)
```

9.3 ХарактеристикаПозиции (PositionMeasure)

ХарактеристикаПозиции (*PositionMeasure*) (пространственное положение) — это характеристика, которая приписывается данному физическому объекту при описании его расположения в пространстве.

Область значений, из которой берутся значения позиции, представляет собой специализацию физической величины согласно определению из онтологии SUMO.

```
# Class: :ХарактеристикаПозиции (:ХарактеристикаПозиции)
SubClassOf(:ХарактеристикаПозиции :Значение)
```

Связь между объектами и значениями измерения положения определяет *ХарактеристикуПозиции*.

```
# Object Property: :находитсяВ (:находитсяВ)
ObjectPropertyDomain(:находитсяВ :Объект)
ObjectPropertyRange(:находитсяВ :ТочкаПозиции)
```

Онтология POS делит *ХарактеристикуПозиции* на точки позиции и области позиции.

```
# Class: :ТочкаПозиции (:ТочкаПозиции)
# Class: :ОбластьПозиции (:ОбластьПозиции)
SubClassOf(:ТочкаПозиции :ХарактеристикаПозиции)
```

ХарактеристикаПозиции всегда определяется в соответствии с одинарной системой координат позиции.

```
# Object Property: :находитсяВ (:находитсяВ)
FunctionalObjectProperty(:находитсяВ)
ObjectPropertyDomain(:находитсяВ :Объект)
ObjectPropertyRange(:находитсяВ :ТочкаПозиции)
```

У эталонного объекта системы координат позиции имеется положение в этой системе координат. Это положение не может измениться.

```
############################
# Named Individuals
############################
# Individual: :ПоложениеЭталонногоОбъекта (:ПоложениеЭталонногоОбъекта)
ClassAssertion(:ТочкаПозиции :ПоложениеЭталонногоОбъекта)
# Individual: :ЭталонныйОбъект (:ЭталонныйОбъект)
ClassAssertion(:Объект :ЭталонныйОбъект)
ObjectPropertyAssertion(:находитсяВ :ЭталонныйОбъект
:ПоложениеЭталонногоОбъекта)
```

Относительное положение объекта по отношению к другому можно определить с помощью *ТрехзначногоПредиката* (*TernaryPredicate*):

```
# Object Property: :относительноеПоложение (:относительноеПоложение)
ObjectPropertyDomain(:относительноеПоложение ObjectSomeValuesFrom(:находитсяВ
:ХарактеристикаПозиции))
ObjectPropertyRange(:относительноеПоложение ObjectSomeValuesFrom(:находитсяВ
:ХарактеристикаПозиции))
```

9.4 ТочкаПозиции (PositionPoint)

ТочкаПозиции (*PositionPoint*) обозначает количественно выраженную позицию объекта в системе координат, спроецированной на физическое пространство.

ТочкаПозиции представляет собой точное указание на позицию данного объекта.

```
# Class: :ТочкаПозиции (:ТочкаПозиции)
SubClassOf(:ТочкаПозиции :ХарактеристикаПозиции)
```

9.5 ПреобразованиеПозиции (PositionTransformation)

ПреобразованиеПозиции (*PositionTransformation*) — это функция, которая сопоставляет одни точки позиции с другими точками позиции (возможно, в другой системе координат).

Преобразования позиции образуют класс *УнарныхФункций* (*UnaryFunction*):

```
# Class: :ПреобразованиеПоложение (:ПреобразованиеПозиции)
EquivalentClasses(:ПреобразованиеПоложение ObjectSomeValuesFrom(:Преобразование
:ТочкаПозиции))
SubClassOf(:ПреобразованиеПоложение :УнарнаяФункция)
# Object Property: :преобразование (:преобразование)
ObjectPropertyDomain(:преобразование :ТочкаПозиции)
ObjectPropertyRange(:преобразование :ТочкаПозиции)
```

Отображение из одной системы координат в другую существует тогда и только тогда, когда существует преобразование, которое отображает все точки одной системы координат в другую.

```
# Object Property: :отображениеСК (:отображениеСК)
SubObjectPropertyOf(:отображениеСК :преобразование)
SymmetricObjectProperty(:отображениеСК)
TransitiveObjectProperty(:отображениеСК)
IrreflexiveObjectProperty(:отображениеСК)
ObjectPropertyDomain(:отображениеСК :ПреобразованиеПозиции)
```

```
ObjectPropertyDomain(:отображениеСК :СистемаКоординат)
ObjectPropertyRange(:отображениеСК :СистемаКоординат)
```

В робототехнике (как и в других дисциплинах) системы координат также связаны между собой через иерархические связи (т. е. «деревья»). Обычно агент выбирает систему координат в качестве глобальной системы отсчета, которая и составляет глобальную систему координат для этого агента. Глобальная система координат может быть выбрана произвольно, без привязки к конкретной системе координат. Локальные системы координат определяются по отношению к глобальной системе координат через иерархические связи. Это понятие иерархии является произвольным, оно определяется агентом.

Если две системы координат связаны между собой через иерархические связи, все точки из одной системы могут быть сопоставлены с точками в другой с помощью преобразований.

Это означает, если у двух систем координат имеется общий родительский узел в иерархическом дереве, тогда между ними имеет место преобразование.

Эти свойства отображений (сопоставлений) между системами координат сохраняются независимо от используемого математического аппарата формализации.

9.6 ОбластьПозиции (PositionRegion)

ОбластьПозиции (*PositionRegion*) определяет не количественное, а качественное расположение объекта.

Она представляет собой абстрактную область в системе координат, перекрывающую физическую пространственную область, занимаемую объектом.

```
# Class: :ТочкаПозиции (:ТочкаПозиции)
# Class: :ОбластьПозиции (:ОбластьПозиции)
SubClassOf(:ТочкаПозиции :ХарактеристикаПозиции)
```

В областях позиции могут находиться точки позиции.

Все точки позиции внутри *ОбластиПозиции* и сама *ОбластьПозиции* находятся в одной системе координат.

```
# Object Property: :входитВ (:входитВ)
ObjectPropertyDomain(:входитВ :ТочкаПозиции)
ObjectPropertyRange(:входитВ :СистемаКоординат)
```

В онтологии POS определены операторы, которые, будучи примененными к объектам, действуют как генераторы *ОбластиПозиции*. Существует множество типов операторов. Например, один оператор может получить точку позиции объекта и сгенерировать *ОбластьПозиции*, соответствующую левой стороне этого объекта.

Когда оператор применяется к списку объектов и системе координат, у всех объектов должна быть позиция в этой системе координат.

Сгенерированная область (т. е. все ее точки) должна находиться в той системе координат, которая используется для ее создания.

Протяженность объекта в системе координат — это минимальная область, содержащая объект в данной системе координат.

ОбластьПозиции связана особыми пространственными связями, которые определены в системах координат, а не в физическом пространстве. Этот стандарт определяет, как минимум, одну связь.

Объект находится в определенной *ОбластиПозиции*, если протяженность объекта, спроецированная на соответствующую систему координат, перекрывается с этой областью позиции.

Относительное положение также может быть определено для *ОбластиПозиции*.

9.7 СистемаКоординатОриентации (OrientationCoordinateSystem)

СистемаКоординатОриентации (*OrientationCoordinateSystem)* — это абстрактная сущность.

```
# Class: :СистемаКоординатОриентации (:СистемаКоординатОриентации)
SubClassOf(:СистемаКоординатОриентации :АбстрактныеПонятия)
```

СистемаКоординатОриентации определена по отношению к одному эталонному объекту.

```
# Object Property: :имеетЭталонный (:имеетЭталонный)
ObjectPropertyDomain(:имеетЭталонный :СистемаКоординатОриентации)
```

```
ObjectPropertyRange(:имеетЭталонный :Объект)
# Individual: :ЭталонныйОбъект (:ЭталонныйОбъект)
ClassAssertion(:Объект :ЭталонныйОбъект)
```

9.8 ХарактеристикаОриентации (OrientationMeasure)

ХарактеристикаОриентации (*OrientationMeasure*) — это измерения, связанные с ориентацией данного физического объекта.

Под ориентацией понимается информация о том, куда направлен объект по отношению к эталонному объекту системы координат ориентации. Например, направление беспилотного летательного аппарата по компасу — это *ХарактеристикаОриентации* в одномерной круговой системе координат ориентации компаса. Эталонным объектом в данном случае является планета Земля.

Область значений, из которого берутся значения ориентации, представляет собой специализацию физической величины.

```
# Class: :ЗначениеОриентации (:ЗначениеОриентации)
SubClassOf(:ЗначениеОриентации :ХарактеристикаОриентации)
```

ХарактеристикаОриентации назначается объектам через связь *«ориентированоНа»*.

```
EquivalentClasses(:ХарактеристикаОриентации ObjectSomeValuesFrom(:ориентировано
На :Объект))
# Object Property: :ориентированоНа (:ориентированоНа)
ObjectPropertyRange(:ориентированоНа :Объект)
```

В онтологии POS *ХарактеристикаОриентации* разделяется на значение ориентации и область ориентации.

```
# Class: :ЗначениеОриентации (:ЗначениеОриентации)
SubClassOf(:ЗначениеОриентации :ХарактеристикаОриентации)
# Class: :ОбластьОриентации (:ОбластьОриентации)
SubClassOf(:ОбластьОриентации :ХарактеристикаОриентации)
```

ХарактеристикаОриентации — это всегда одинарная *СистемаКоординатОриентации* (*OrientationCoordinateSystem)*.

```
# Class: :ХарактеристикаОриентации (:ХарактеристикаОриентации)
EquivalentClasses(:ХарактеристикаОриентации ObjectSomeValuesFrom(:входитВ
:СистемаКоординатОриентации))
# Object Property: :входитВ (:входитВ)
FunctionalObjectProperty(:входитВ)
ObjectPropertyDomain(:входитВ :ХарактеристикаОриентации)
ObjectPropertyRange(:входитВ :СистемаКоординатОриентации)
```

Можно привести множество примеров *ХарактеристикиОриентации*: от простых углов ориентации до векторов и других математических представлений. Тип *СистемыКоординатОриентации* определяет, какие значения ориентации имеются у системы координат. Например, можно специализировать систему координат ориентации таким образом, чтобы определить двумерную векторную систему ориентации, в которой значениями ориентации являются двумерные единичные векторы.

9.9 ЗначениеОриентации (OrientationValue)

ЗначениеОриентации (*OrientationValue*) обозначает количественные значения ориентации данного объекта в системе координат ориентации.

Значение ориентации — это точное указание ориентации данного объекта.

```
# Class: :ЗначениеОриентации (:ЗначениеОриентации)
SubClassOf(:ЗначениеОриентации :ХарактеристикаОриентации)
```

9.10 ПреобразованиеОриентации (OrientationTransformation)

ПреобразованиеОриентации (*OrientationTransformation*) — это функция, которая отображает *ЗначениеОриентации* на *ЗначениеОриентации* (возможно в другой *СистемеКоординатОриентации*).

```
# Class: :ПреобразованиеОриентации (:ПреобразованиеОриентации)
SubClassOf(:ПреобразованиеОриентации :УнарнаяФункция)
# Object Property: :преобразованиеОриентации (:преобразованиеОриентации)
SubObjectPropertyOf(:преобразованиеОриентации :преобразование)
ObjectPropertyDomain(:преобразованиеОриентации :ЗначениеОриентации)
ObjectPropertyRange(:преобразованиеОриентации :ЗначениеОриентации)
```

Отображение из одной системы координат ориентации на другую существует тогда и только тогда, когда существует преобразование, которое отображает все значения ориентации из первой системы координат на вторую.

Как и в случае с *СистемойКоординатПозиции*, если две *СистемыКоординатОриентации* связаны иерархически, то можно отобразить все точки из одной системы координат в другую с помощью преобразований.

Это означает, что, если у двух систем координат имеется общий родительский узел в иерархическом дереве, тогда между ними имеет место преобразование.

9.11 ОбластьОриентации (OrientationRegion)

ОбластьОриентации (OrientationRegion) — это абстрактная область, используемая для представления не количественной, а качественной информации об ориентации в *СистемеКоординатОриентации*.

ОбластьОриентации аналогична областям позиции в том смысле, что она представляет общую, обычно неточную информацию об ориентации. Например, предложение «робот движется на запад» указывает на то, что *ОбластьОриентации* робота в *СистемеКоординатОриентации* компаса — запад. В этом случае запад — это область в системе координат в том смысле, что она охватывает все значения ориентации, которые обычно интерпретируются как «направление на запад».

Пространство ориентации, показанное на рисунке 10, представляет собой одномерное круговое пространство. Оно разделено на четыре сегмента, каждый из которых соответствует *ОбластиОриентации*. Сегмент, который выделен на рисунке, соответствует западной *ОбластиОриентации*.

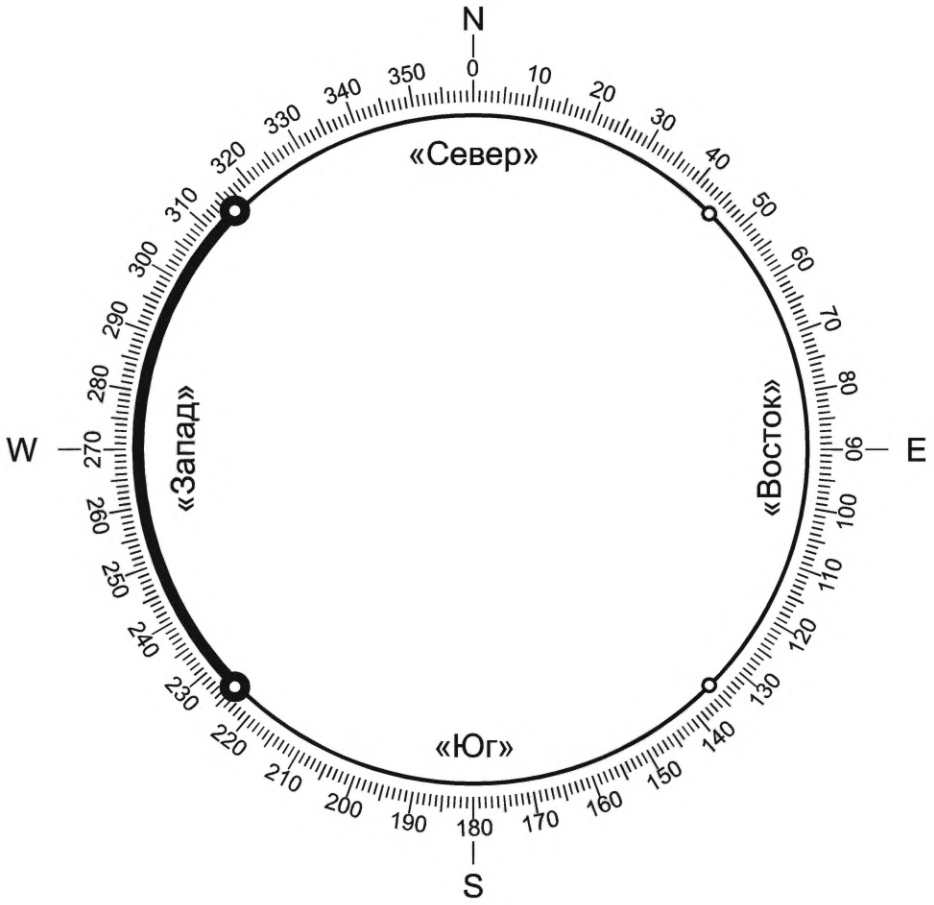


Рисунок 10 — Значения ориентации и области ориентации в системе координат компаса

ОбластьОриентации определяется *ЗначениемОриентации*:

```
# Class: :ОбластьОриентации (:ОбластьОриентации)
SubClassOf(:ОбластьОриентации :ХарактеристикаОриентации)
```

Значения ориентации расположены внутри *ОбластиОриентации*.

Все значения ориентации в *ОбластиОриентации* и сама *ОбластьОриентации* находятся в одной и той же системе координат.

```
# Class: :ЗначениеОриентации (:ЗначениеОриентации)
SubClassOf(:ЗначениеОриентации :ХарактеристикаОриентации)
# Class: :ОбластьОриентации (:ОбластьОриентации)
SubClassOf(:ОбластьОриентации :ХарактеристикаОриентации)
# Class: :ХарактеристикаОриентации (:ХарактеристикаОриентации)
EquivalentClasses(:ХарактеристикаОриентации ObjectSomeValuesFrom(:ОриентированоНа :Объект))
EquivalentClasses(:ХарактеристикаОриентации ObjectExactCardinality(1 :ВходитВ :СистемаКоординатОриентации))
SubClassOf(:ХарактеристикаОриентации :Значение)
```

В онтологии POS определены операторы, которые, будучи примененными к объектам, действуют как генератор *ОбластиОриентации*. Существует множество типов операторов. Например, один оператор может получить значение ориентации объекта и сгенерировать область ориентации, которая соответствует северу для этого объекта.

Относительная ориентация также может быть определена для *ОбластиОриентации*.

9.12 ПространственноеРасположение (Pose)

ПространственноеРасположение (*Pose*) некоего объекта — это величина, содержащая ориентацию и позицию.

```
# Class: :ХарактеристикаПространственногоРасположения
(:ХарактеристикаПространственногоРасположения)
SubClassOf(:ХарактеристикаПространственногоРасположения :Значение)
# Class: :ХарактеристикаОриентации (:ХарактеристикаОриентации)
SubClassOf(:ХарактеристикаОриентации :Значение)
# Class: :ТочкаПозиции (:ТочкаПозиции)
SubClassOf(:ТочкаПозиции :ХарактеристикаПространственногоРасположения)
# Object Property: :пространственноеРасположениеОриентация
(:пространственноеРасположениеОриентация)
SubObjectPropertyOf(:пространственноеРасположениеОриентация
owl:topObjectProperty)
FunctionalObjectProperty(:пространственноеРасположениеОриентация)
ObjectPropertyDomain(:пространственноеРасположениеОриентация :Объект)
ObjectPropertyRange(:пространственноеРасположениеОриентация
:ХарактеристикаОриентации)
ObjectPropertyRange(:пространственноеРасположениеОриентация
:ХарактеристикаПространственногоРасположения)
# Object Property: :пространственноеРасположениеПозиция
(:пространственноеРасположениеПозиция)
FunctionalObjectProperty(:пространственноеРасположениеПозиция)
ObjectPropertyDomain(:пространственноеРасположениеПозиция :Объект)
ObjectPropertyRange(:пространственноеРасположениеПозиция
:ХарактеристикаПространственногоРасположения)
```

ХарактеристикаПространственногоРасположения (*PoseMeasure*), имеющая *ЗначениеОриентации* (*OrientationValue*) и *ТочкуПозиции* (*PositionPoint*) называется *КоличественнойОценкойПространственногоРасположения* (*QuantitativePose*).

```
# Class: :КоличественнаяОценкаПространственногоРасположения
(:КоличественнаяОценкаПространственногоРасположения)
EquivalentClasses(:КоличественнаяОценкаПространственногоРасположения
ObjectIntersectionOf(ObjectSomeValuesFrom(:пространственноеРасположениеОриентац
ия :ЗначениеОриентации) ObjectSomeValuesFrom(:пространственноеРасположениеПозиц
ия :ТочкаПозиции)))
SubClassOf(:КоличественнаяОценкаПространственногоРасположения
:ХарактеристикаПространственногоРасположения)
```

9.13 ПреобразованиеПространственногоРасположения (PoseTransformation)

Между количественными оценками пространственного расположения можно определить функцию *преобразования* (*transformations*). Данные преобразования образуют класс унарных функций:

```
# Class: : КоличественнаяОценкаПреобразованияПространственногоРасположения
(:КоличественнаяОценкаПреобразованияПространственногоРасположения)
EquivalentClasses(:КоличественнаяОценкаПреобразованияПространственного
Расположения ObjectSomeValuesFrom(:преобразованиеПространственногоРасположения
:КоличественнаяОценкаПространственногоРасположения))
SubClassOf(:КоличественнаяОценкаПреобразованияПространственногоРасположения
:УнарнаяФункция)
# Object Property: :преобразованиеПространственногоРасположения
(:преобразованиеПространственногоРасположения)
ObjectPropertyDomain(:преобразованиеПространственногоРасположения
:КоличественнаяОценкаПространственногоРасположения)
ObjectPropertyRange(:преобразованиеПространственногоРасположения
:КоличественнаяОценкаПространственногоРасположения)
```

КоличественнаяОценкаПреобразованияПространственногоРасположения (*QuantitativePoseTransformation*) означает оценку изменения значений местоположения и ориентации, характеризующих пространственное расположение объекта.

УДК 621.865.8:629.78:006.354

ОКС 25.040.30
35.060

Ключевые слова: роботы, робототехнические устройства, онтологии, робототехника, основные положения, термины и определения

Редактор *Н.А. Аргунова*
Технический редактор *И.Е. Черепкова*
Корректор *И.А. Королева*
Компьютерная верстка *М.В. Малеевой*

Сдано в набор 25.09.2023. Подписано в печать 28.09.2023. Формат 60×84⅛. Гарнитура Ариал.
Усл. печ. л. 4,65. Уч.-изд. л. 4,18.

Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

Создано в единичном исполнении в ФГБУ «Институт стандартизации»
для комплектования Федерального информационного фонда стандартов,
117418 Москва, Нахимовский пр-т, д. 31, к. 2.

gost.e-ecolog.ru gost.e-ecolog.ru

gost.e-ecolog.ru