МЕЖДУНАРОДНЫЙ СОЮЗ ЭЛЕКТРОСВЯЗИ

МСЭ-Т X.1142

(06/2006)

СЕКТОР СТАНДАРТИЗАЦИИ
ЭЛЕКТРОСВЯЗИ МСЭ

СЕРИЯ X: СЕТИ ПЕРЕДАЧИ ДАННЫХ, ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ И БЕЗОПАСНОСТЬ

Безопасность электросвязи

Расширяемый язык разметки контроля доступа (XACML 2.0)

Рекомендация МСЭ-Т X.1142

РЕКОМЕНДАЦИИ МСЭ-Т СЕРИИ X

СЕТИ ПЕРЕДАЧИ ДАННЫХ, ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ И БЕЗОПАСНОСТЬ

СЕТИ ПЕРЕДАЧИ ДАННЫХ ОБЩЕГО ПОЛЬЗОВАНИЯ	
Службы и услуги	X.1–X.19
Интерфейсы	X.20–X.49
Передача, сигнализация и коммутация	X.50–X.89
Сетевые аспекты	X.90–X.149
Техническое обслуживание	X.150–X.179
Административные предписания	X.180–X.199
ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ	
Модель и обозначение	X.200–X.209
Определения служб	X.210–X.219
Спецификации протоколов с установлением соединений	X.220–X.229
Спецификации протоколов без установления соединений	X.230–X.239
Проформы PICS	X.240–X.259
Идентификация протоколов	X.260–X.269
Протоколы обеспечения безопасности	X.270–X.279
Управляемые объекты уровня	X.280–X.289
Испытание на соответствие	X.290–X.299
ВЗАИМОДЕЙСТВИЕ МЕЖДУ СЕТЯМИ	
Общие положения	X.300–X.349
Спутниковые системы передачи данных	X.350–X.369
Сети, основанные на протоколе Интернет	X.370–X.379
СИСТЕМЫ ОБРАБОТКИ СООБЩЕНИЙ	X.400–X.499
СПРАВОЧНИК	X.500–X.599
ОРГАНИЗАЦИЯ СЕТИ ВОС И СИСТЕМНЫЕ АСПЕКТЫ	
Организация сети	X.600–X.629
Эффективность	X.630–X.639
Качество обслуживания	X.640–X.649
Наименование, адресация и регистрация	X.650–X.679
Абстрактно-синтаксическая нотация 1 (ASN.1)	X.680–X.699
УПРАВЛЕНИЕ В ВОС	
Структура и архитектура управления системами	X.700–X.709
Служба и протокол связи для общего управления	X.710–X.719
Структура управляющей информации	X.720–X.729
Функции общего управления и функции ODMA	X.730–X.799
БЕЗОПАСНОСТЬ	X.800–X.849
ПРИЛОЖЕНИЯ ВОС	
Фиксация, параллельность и восстановление	X.850–X.859
Обработка транзакций	X.860–X.879
Удаленные операции	X.880–X.889
Общие приложения ASN.1	X.890–X.899
ОТКРЫТАЯ РАСПРЕДЕЛЕННАЯ ОБРАБОТКА	X.900–X.999
БЕЗОПАСНОСТЬ ИНФОРМАЦИИ И СЕТЕЙ	**X.1000–**

Для получения более подробной информации просьба обращаться к перечню Рекомендаций МСЭ-Т.

Рекомендация МСЭ-Т X.1142

Расширяемый язык разметки контроля доступа (XACML 2.0)

Резюме

XACML (Расширяемый язык разметки контроля доступа) – это словарь XML (Расширяемый язык разметки) для выражения стратегии контроля доступом. Контроль доступа заключается в принятии решения о разрешении запрашиваемого доступа к ресурсу и в осуществлении этого решения. В данной Рекомендации определяется основа XACML, включая синтаксис языка, модели, контекст с моделью языка стратегии, правила синтаксиса и обработки информации. В данной Рекомендации устанавливается профиль основного и иерархического контроля доступа на ролевой основе для языка XACML. Устанавливается профиль множества ресурсов языка XACML и профиль SAML 2.0 языка XACML. Для усовершенствования безопасности при обмене стратегией на основе XACML в данной Рекомендации также устанавливается профиль цифровой подписи XML языка XACML для организации защиты данных. Профиль секретности устанавливается с целью предоставления руководящих указаний для реализаций.

Данная Рекомендация с технической точки зрения эквивалентна и совместима со стандартом OASIS XACML 2.0.

Источник

Рекомендация МСЭ-Т X.1142 утверждена 13 июня 2006 г. 17-й Исследовательской комиссией МСЭ-Т (2005–2008 гг.) в соответствии с процедурой, изложенной в Рекомендации МСЭ-Т A.8.

ПРЕДИСЛОВИЕ

Международный союз электросвязи (МСЭ) является специализированным учреждением Организации Объединенных Наций в области электросвязи и информационно-коммуникационных технологий (ИКТ). Сектор стандартизации электросвязи МСЭ (МСЭ-Т) – постоянный орган МСЭ. МСЭ-Т отвечает за изучение технических, эксплуатационных и тарифных вопросов и за выпуск Рекомендаций по ним с целью стандартизации электросвязи на всемирной основе.

На Всемирной ассамблее по стандартизации электросвязи (ВАСЭ), которая проводится каждые четыре года, определяются темы для изучения Исследовательскими комиссиями МСЭ-Т, которые, в свою очередь, вырабатывают Рекомендации по этим темам.

Утверждение Рекомендаций МСЭ-Т осуществляется в соответствии с процедурой, изложенной в Резолюции 1 ВАСЭ.

В некоторых областях информационных технологий, которые входят в компетенцию МСЭ-Т, необходимые стандарты разрабатываются на основе сотрудничества с ИСО и МЭК.

ПРИМЕЧАНИЕ

В настоящей Рекомендации термин "администрация" используется для краткости и обозначает как администрацию электросвязи, так и признанную эксплуатационную организацию.

Соблюдение положений данной Рекомендации осуществляется на добровольной основе. Однако данная Рекомендация может содержать некоторые обязательные положения (например, для обеспечения функциональной совместимости или возможности применения), и в таком случае соблюдение Рекомендации достигается при выполнении всех указанных положений. Для выражения требований используются слова "следует", "должен" ("shall") или некоторые другие обязывающие выражения, такие как "обязан" ("must"), а также их отрицательные формы. Употребление таких слов не означает, что от какой-либо стороны требуется соблюдение положений данной Рекомендации.

ПРАВА ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

МСЭ обращает внимание на вероятность того, что практическое применение или выполнение настоящей Рекомендации может включать использование заявленного права интеллектуальной собственности. МСЭ не занимает какую бы то ни было позицию относительно подтверждения, действительности или применимости заявленных прав интеллектуальной собственности, независимо от того, доказываются ли такие права членами МСЭ или другими сторонами, не относящимися к процессу разработки Рекомендации.

На момент утверждения настоящей Рекомендации МСЭ не получил извещение об интеллектуальной собственности, защищенной патентами, которые могут потребоваться для выполнения настоящей Рекомендации. Однако те, кто будет применять Рекомендацию, должны иметь в виду, что вышесказанное может не отражать самую последнюю информацию, и поэтому им настоятельно рекомендуется обращаться к патентной базе данных БСЭ по адресу: http://www.itu.int/ITU-T/ipr/.

СОДЕРЖАНИЕ

Стр.

Рекомендация МСЭ-Т X.1142

Расширяемый язык разметки контроля доступа (XACML 2.0)

1 Область применения

В данной Рекомендации определен расширяемый язык разметки контроля доступа (XACML) версия 2.0. В ней определен общий язык для выражения стратегии безопасности. Мотивация, стоящая за созданием XACML – это развитие языка стратегии на базе XML, который может использоваться для:

- предоставления метода для объединения отдельных правил и стратегий в единый набор стратегий, применимый к конкретному запросу о принятии решения.
- предоставления метода для гибкого определения процедуры, с помощью которой объединяются правила и стратегии.
- предоставления метода, для того чтобы иметь дело с множеством субъектов, действующих в разных областях.
- предоставления метода для обоснования решения по вопросу авторизации на основе атрибутов субъекта и ресурса.
- предоставления метода, для того чтобы иметь дело с многозначными атрибутами.
- предоставления метода для обоснования решения по вопросу авторизации на основе содержания ресурса информации.
- предоставления набора логических и математических операторов по атрибутам субъекта, ресурса и среды.
- предоставления метода для обработки распределенного набора компонентов стратегии, наряду с обобщением метода для локализации, поиска и аутентификации компонентов стратегии.
- предоставления метода для быстрой идентификации стратегии, которая применяется к заданному действию, основанному на значениях атрибутов субъектов, ресурсов и действий.
- предоставления уровня абстракции, который абстрагирует автора стратегии от подробностей среды применения.
- предоставления метода для устанавливания набора действий, которые должны быть выполнены совместно с осуществлением стратегии.

Решения XACML для каждого из этих требований находятся в данной Рекомендации. В частности, в пункте 7 разрабатывается основной язык XACML, включая модели, модель языка стратегии, правила синтаксиса стратегии и обработки информации XACML. В пункте 8 разрабатывается профиль основного и иерархического контроля доступа на ролевой основе (RBAC) языка XACML. В пункте 9 разрабатывается профиль множества ресурсов XACML. В пункте 10 обсуждаются технологии для организации защиты связи XACML посредством развития профиля SAML 2.0 языка XACML. Пункт 11 построен на основе пункта 10 с помощью разработки профиля цифровой подписи XML для XACML. В пункте 12 обсуждается объединение профилей XACML, как разработано в пунктах с 7 до 11, с помощью разработки профиля иерархических ресурсов XACML. Темы секретности обсуждаются в пункте 13.

2 Справочная литература

Указанные ниже Рекомендации и другая справочная литература содержит положения, которые путем ссылки на них в данном тексте составляют положения настоящей Рекомендации. На момент публикации указанные издания были действующими. Все Рекомендации и другая справочная литература может подвергаться пересмотру; поэтому сторонам для соглашений, основанных на данной Рекомендации, предлагается изучить возможность применения последнего издания Рекомендаций и другой справочной литературы, перечисленной ниже. Бюро стандартизации электросвязи МСЭ ведет список действующих в настоящее время Рекомендаций МСЭ-Т. IETF ведет список RFC (Рабочие предложения), вместе с теми, которые были отменены более поздними редакциями RFC. W3C (Консорциум World Wide Web) ведет список последних Рекомендаций и других публикаций.

- ITU-T Recommendation X.811 (1995) | ISO/IEC 10181-2:1996, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Authentication framework*.
- ITU-T Recommendation X.812 (1995) | ISO/IEC 10181-3:1996, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Access control framework*.
- ITU-T Recommendation X.1141 (2006), *Security Assertion Markup Language (SAML 2.0)*.
- IETF RFC 822 (1982), *Standard for the Format of ARPA Internet Text Messages*.
- IETF RFC 2119 (1997), *Key words for use in RFCs to Indicate Requirement Levels*.
- IETF RFC 2253 (1997), *Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names*.
- IETF RFC 2256 (1997), *A Summary of the X.500 (96) User Schema for use with LDAPv3*.

– IETF RFC 2396 (1998), *Uniform Resource Identifiers (URI): Generic Syntax*.

– IETF RFC 2732 (1999), *Format for Literal IPv6 Addresses in URL's*.

– IETF RFC 2821 (2001), *Simple Mail Transfer Protocol*.

– IETF RFC 3280 (2002), *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*.

– W3C Canonicalization:2002, *Exclusive XML Canonicalization Version 1.0*, W3C Recommendation, Copyright © [18 July 2002] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/xml-exc-c14n/.

– W3C Datatypes:2001, *XML Schema Part 2: Datatypes*, W3C Recommendation, Copyright © [2 May 2001] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/.

– W3C Encryption:2002, *XML Encryption Syntax and Processing*, W3C Recommendation, Copyright © [10 December 2002] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/.

– W3C MathML:2003, *Mathematical Markup Language (MathML)*, *Version 2.0*, W3C Recommendation, Copyright © [21 October 2003] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/2003/REC-MathML2-20031021/.

– W3C Signature:2002, *XML-Signature Syntax and Processing*, W3C Recommendation, Copyright © [12 February 2002] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/.

– W3C XML:2004, *Extensible Markup Language (XML) 1.0 (Third Edition)*, W3C Recommendation, Copyright © [4 February 2004] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/REC-xml/.

– W3C XPATH:1999, *XML Path Language, Version 1.0,* W3C Recommendation, Copyright © [16 November 1999] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/1999/REC-xpath-19991116.

– W3C XSLT:1999, *XSL Transformations (XSLT) Version 1.0*, W3C Recommendation, Copyright © [16 November 1999] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/1999/REC-xslt-19991116.

ПРИМЕЧАНИЕ. – Ссылка на документ в рамках этой Рекомендации не даёт ему, как отдельному документу, статуса Рекомендации.

3 Определения

Для целей данной Рекомендации применяют следующие определения.

3.1 Заимствованные определения

3.1.1 В данной Рекомендации используются следующие термины, определенные в Рекомендации МСЭ-Т X.811:

a) Принцип.

3.1.2 В данной Рекомендации используются следующие термины, определенные в Рекомендации МСЭ-Т X.812:

a) Информация контроля доступа;

b) Пользователь.

3.1.3 В данной Рекомендации используются следующие термины, определенные в W3C Web Services Glossary:

a) Пространство имен;

b) Схема XML.

3.1.4 В данной Рекомендации используются следующие термины, определенные в IETF RFC 2828:

a) Доступ;

b) Контроль доступа;

c) Пункт управления стратегией;

d) Пункт выбора стратегии;

e) Пункт осуществления стратегии;

f) Архитектура безопасности;

g) Стратегия безопасности;

h) Служба безопасности.

3.1.5 В данной Рекомендации используются следующие термины, определенные в IETF RFC 2396:

a) Унифицированный идентификатор ресурса (URI);

b) Ссылка URI.

3.1.6 В данной Рекомендации используются следующие термины, определенные в W3C XML Signature:

a) Объект данных.

3.2 Дополнительные определения

3.2.1 **доступ**: Выполнение действия.

3.2.2 **контроль доступа**: Контролирование доступа в соответствии со стратегией.

3.2.3 **действие**: Операция над ресурсом.

3.2.4 **применяемая стратегия**: Набор стратегий и стратегические наборы, которые управляют доступом для конкретного запроса о принятии решения.

3.2.5 **атрибут**: Характеристика субъекта, ресурса, действия или среды, на которую могут ссылаться в предикате или цели**.**

3.2.6 **орган атрибута (AA)**: Объект, который связывает атрибуты с идентичностью. Такое связывание можно выразить с помощью использования утверждения атрибута SAML с органом атрибута, как запрашивающей стороной.

3.2.7 **решение об авторизации**: Результат оценки применяемой стратегии, возвращается с помощью PDP (Пункт выбора стратегии) к PEP (Пункт осуществления стратегии). Функция, которая оценивается как "Permit", "Deny", "Indeterminate" или "NotApplicable", и (необязательно) набор обязательств.

3.2.8 **мешок (множество с повторяющимися элементами)**: Неупорядоченный набор значений, среди которых могут быть повторяющиеся значения.

3.2.9 **условие**: Выражение предикат. Функция, которая оценивается как "True", "False" или "Indeterminate".

3.2.10 **конъюнктивная последовательность**: Последовательность предикат, объединенных с использованием операции логического "AND".

3.2.11 **контекст**: Каноническое представление запроса о принятии решения и решения об авторизации.

3.2.12 **обработчик контекста**: Системный объект, который преобразует запросы о принятии решения, выраженные в присущем данной системе формате, в каноническую форму запроса XACML и преобразует решения об авторизации, выраженные в канонической форме, в присущий данной системе формат.

3.2.13 **хранитель**: Объект, которому вверяется лично идентифицируемая информация.

3.2.14 **объект данных**: Относится к подписываемому цифровому объекту. На объект данных ссылаются внутри элемента `<Reference>`, используя URI.

3.2.15 **решение**: Результат оценки правила, стратегии или стратегического набора.

3.2.16 **запрос о принятии решения**: Запрос с помощью PEP к PDP для представления решения об авторизации.

3.2.17 **дизъюнктивная последовательность**: Последовательность предикат, объединенная с использованием операции логического 'OR'.

3.2.18 **эффект**: Намеченное следствие выполненного правила (либо "Permit", либо "Deny").

3.2.19 **среда**: Набор атрибутов, которые относятся к решению об авторизации и независимы от конкретного предмета, ресурса или действия.

3.2.20 **стратегия HasPrivilegesOfRole**: Необязательный тип `<Policy>`, который может быть включен в разрешение `<PolicySet>`, чтобы позволить поддержку запросов о том "есть ли привилегии" особой роли у субъекта.

3.2.21 иерархический ресурс: Ресурс, организованный как дерево или лес (направленные непериодические графы) индивидуальных ресурсов, называемых узлами.

3.2.22 младшая роль: В ролевой иерархии роль A является *junior* по отношению к роли B, если роль B наследует все разрешения, связанные с ролью A.

3.2.23 многоролевые разрешения: Набор разрешений, для которых пользователь должен выполнять одновременно более одной роли, для того чтобы получить доступ.

3.2.24 именованный атрибут: Особый экземпляр атрибута, определенный с помощью имени и типа, идентичности держателя атрибута (который может быть таких типов: субъект, ресурс, действие или среда) и (необязательно) идентичности выпускающего органа.

3.2.25 узел: Индивидуальный ресурс, являющийся частью иерархического ресурса.

3.2.26 обязательство: Операция, установленная в стратегии или стратегическом наборе, которая должна выполняться с помощью PEP вместе с осуществлением решения об авторизации.

3.2.27 владелец: Субъект лично идентифицируемой информации.

3.2.28 разрешение: Способность или право выполнять некоторые действия над каким-то ресурсом, возможно только при точно установленных условиях.

3.2.29 разрешение `<policy set>` (PPS): `<PolicySet>`, в котором содержатся фактические разрешения, связанные с заданной ролью.

3.2.30 информационный пункт стратегии (PIP): Системный объект, который ведет себя, как источник значений атрибута.

3.2.31 набор стратегий: Набор стратегий, другие стратегические наборы, алгоритм объединения стратегий и (необязательно) набор обязательств. Может быть компонентом другого набора стратегий.

3.2.32 предикат: Утверждение об атрибутах, справедливость которого может быть установлена.

3.2.33 ресурс: Данные, услуга или системный компонент.

3.2.34 роль: Рабочая функция внутри контекста организации, в которой имеется связанная семантика, относящаяся к полномочию и ответственности, предоставленная пользователю, связанному с этой ролью.

3.2.35 контроль доступа на ролевой основе (RBAC): Модель для контролирования доступа к ресурсам, в которой разрешенные действия над ресурсом идентифицируются ролями, а не отдельными идентичностями субъектов.

3.2.36 орган ролевых разрешений: Объект, который присваивает ролевые атрибуты и значения пользователям, или разрешает ролевые атрибуты и значения в течение сеанса пользователя.

3.2.37 роль `<PolicySet>` (RPS): `<PolicySet>`, который связывает держателей атрибута и значения заданной роли с Разрешением `<PolicySet>`, в котором содержатся фактические разрешения, связанные с заданной ролью.

3.2.38 старшая роль: В ролевой иерархии роль A является *senior* по отношению к роли B, если роль A наследует все разрешения, связанные с ролью B.

3.2.39 правило: Цель, эффект и условие. Компонент стратегии.

3.2.40 алгоритм объединения правил: Процедура для объединения решений, исходящих от множества правил.

3.2.41 субъект: Действующий объект, на атрибуты которого можно ссылаться посредством предиката.

3.2.42 цель: Набор запросов о принятии решения, идентифицированный определениями для ресурса, субъекта и действия, который предназначен для оценки с помощью правила, стратегии или стратегического набора.

3.2.43 типовая унификация: Метод, с помощью которого два типовых выражения "унифицируются". Типовые выражения подбираются в соответствии с их структурой. Если типовая переменная появляется в одном выражении, то потом она "унифицируется", для того чтобы представлять соответствующий структурный элемент другого выражения, будь то другая переменная или подвыражение. Все присвоения переменной должны оставаться согласованными в обеих структурах. Унификация не удается, если два выражения нельзя выровнять либо из-за несхожей структуры, либо из-за конфликтов экземпляров, если например, переменная должна представлять как "**xs:string**", так и "**xs:integer**".

4 Сокращения

Для целей данной Рекомендации, применяют следующие сокращения:

AA	Attribute Authority	Орган атрибута
ASP	Application Service Provider	Поставщик услуг по аренде приложений
CA	Certification Authority	Бюро сертификации
CMP	Certificate Management Protocol	Протокол управления сертификатом
CRL	Certificate Revocation List	Список аннулированных сертификатов
ECP	Enhanced Client/Proxy	Усиленный клиент/прокси
HTTP	Hypertext Transfer Protocol	Протокол передачи гипертекстовых файлов
ID	IDentifier	Идентификатор
IPSEC	IP SECurity protocol	Интернет-протокол безопасности
LDAP	Lightweight Directory Access Protocol	Упрощенный протокол доступа к Справочнику
PAP	Policy Administration Point	Пункт административного управления стратегией
PDP	Policy Decision Point	Сервер стратегии, пункт выбора стратегии
PEP	Policy Enforcement Point	Клиент сервера стратегии, пункт осуществления стратегии
PIP	Policy Information Point	Информационный пункт стратегии
PKI	Public-Key Infrastructure	Инфраструктура открытого ключа
POP	Proof of Possession	Доказательство права владения
PPS	Permission `<Policy Set>`	Разрешение `<Policy Set>`
RA	Registration Authority	Регистрационный орган
RBAC	Role Based Access Control	Контроль доступа на ролевой основе
RPS	Role `<PolicySet>`	Роль `<PolicySet>`
RSA	Rivest, Shamir, Adleman (public key algorithm)	Райвест, Шамир, Адлеман (алгоритм открытого ключа)
SAML	Security Assertion Markup Language	Язык разметки для систем обеспечения безопасности
SP	Service Provider	Поставщик услуг
SSO	Single Sign On	Единственная подпись
TLS	Transport Layer Security	Протокол защиты транспортного уровня
URI	Uniform Resource Identifier	Унифицированный идентификатор ресурса
URN	Uniform Resource Name	Имя унифицированного ресурса
XML	eXtensible Markup Language	Расширяемый язык разметки, язык XML
XPath	XML Path Language	Язык ветви XML
XSLT	eXtensible Stylesheet Language	Расширяемый язык таблицы стилей

5 Соглашения о терминах

В данной Рекомендации используются ключевые слова "обязан", "запрещается", "требуется", "должен", "не должен", "следует", "не следует", "рекомендуется", "можно" и "необязательный". В данной Рекомендации эти термины должны интерпретироваться, как описано в RFC 2119.

В данной Рекомендации фраза "Нормативный, но необязательный" означает, что описанная функциональная возможность является необязательной для соответствующих реализаций XACML, но если эта функциональная возможность заявлена, как поддерживающаяся в соответствии с данным профилем, тогда она должна поддерживаться описанным способом.

В описаниях синтаксиса, элементы в угловых скобках ("<", ">") должны быть заменены на подходящие значения, в квадратных скобках ("[", "]") заключены необязательные элементы, элементы в кавычках являются буквенными компонентами, а "*" указывает на то, что предыдущий элемент может встречаться ноль или большее количество раз.

6 Обзор

В связи с эффектом масштаба поставщики стандартизованного компьютерного оборудования пришли к разработке продуктов с очень обобщенной функциональностью, для того чтобы их можно было использовать в самом широком диапазоне ситуаций. У этих "новеньких, с иголочки" продуктов имеются максимальные возможные привилегии для организации доступа к данным и выполнения программ, для того чтобы их можно было использовать в как можно большем количестве прикладных сред, включая среды с самыми разрешающими стратегиями безопасности. В самом общем случае со сравнительно ограничивающей стратегией безопасности, привилегии, присущие стандартному компьютерному оборудованию, должны быть ограничены конфигурацией.

У стратегии безопасности большого предприятия имеется много элементов и много пунктов осуществления. Элементами стратегии может управлять отдел информационных систем, отдел кадров, юридический отдел и отдел финансов. Стратегия также осуществляется с помощью сети экстранет, почтой, системами WAN (территориально-распределенная сеть) и системами удаленного доступа; стандартизованным компьютерным оборудованием, которое, по сути, реализует разрешающую стратегию безопасности. В текущей практике управление конфигурацией каждого пункта осуществления должно осуществляться независимо, с целью реализации стратегии безопасности с наибольшей точностью. Следовательно, предложение об изменении стратегии безопасности является дорогостоящим и ненадежным. И практически невозможно достичь общей точки зрения охраны фактически по всему предприятию для осуществления стратегии. В то же время, присутствует все возрастающее давление на корпоративных и государственных исполнителей со стороны потребителей, акционеров и инспекторов для демонстрации "передового опыта" по вопросу защиты информационных активов предприятия и его заказчиков.

По этим причинам ощущается настоятельная потребность в создании общего языка для выражения стратегии безопасности. Если такой общий язык стратегии будет реализован по всему предприятию, то он даст возможность этому предприятию управлять пунктами осуществления всех элементов этой стратегии безопасности во всех компонентах его информационных систем. Управляющая стратегия безопасности может включать в себя некоторые или все следующие шаги: написание, рецензирование, проверку, одобрение, выпуск, объединение, проведение анализа, модификацию, отзыв, поиск и стратегию осуществления.

7 Основной XACML

В этом пункте разрабатываются фундаментальные основы языка XACML, включая общие требования к стратегии, модели, общее содержание, синтаксис стратегии, и приводятся примеры.

7.1 Базовая информация

Этот пункт является информативным.

XML является естественным выбором в качестве основы для общего языка стратегии безопасности, из-за легкости, с которой его синтаксис и семантика могут быть расширены, чтобы приспособить уникальные требования приложения, и благодаря той широкой поддержке, которой он пользуется у основных поставщиков стандартизованного компьютерного оборудования и инструментов.

7.1.1 Требования

Основными требованиями языка стратегии для выражения стратегии безопасности информационной системы являются:

- Предоставления метода для объединения отдельных правил и стратегий в единый набор стратегий, применимый к запросу частного определения.
- Предоставления метода для гибкого определения процедуры, с помощью которой объединяются правила и стратегии.
- Предоставления метода, для того чтобы иметь дело с множеством субъектов, действующих в разных областях.
- Предоставления метода для обоснования решения по вопросу авторизации на основе атрибутов субъекта и ресурса.
- Предоставления метода, для того чтобы иметь дело с многозначными атрибутами.
- Предоставления метода для обоснования решения по вопросу авторизации на основе содержания ресурса информации.
- Предоставления набора логических и математических операторов по атрибутам предмета, ресурса и среды.
- Предоставления метода для обработки распределенного набора компонентов стратегии, наряду с обобщением метода для локализации, поиска и аутентификации компонентов стратегии.
- Предоставления метода для быстрой идентификации стратегии, которая применяется к заданному действию, основанному на значениях атрибутов субъектов, ресурсов и действий.
- Предоставления уровня абстракции, который абстрагирует автора стратегии от подробностей среды применения.
- Предоставления метода для установления набора действий, которые должны быть выполнены одновременно с осуществлением стратегии.

Мотивация, стоящая за созданием XACML – это выражение этих хорошо обоснованных идей в области стратегии доступа-контроля с использованием расширения языка XML. Решения XACML для каждого из этих требований обсуждаются в следующих пунктах.

7.1.2 Объединение правил и стратегий

Завершенная стратегия, применимая к конкретному запросу о принятии решения, может быть составлена из нескольких индивидуальных правил и стратегий. Например, в приложении о неприкосновенности личной жизни владелец личной информации может определять некоторые аспекты стратегии разглашения, в то время как предприятие, которое является хранителем информации, может определять некоторые другие аспекты. Для получения решения об авторизации должна иметься возможность объединения двух отдельных стратегий для образования единой стратегии, применимой к этому запросу.

В XACML определены три элемента стратегии наивысшего уровня: `<Rule>`, `<Policy>` и `<PolicySet>`. В элементе `<Rule>` содержится логическое выражение, которое может быть оценено изолированно, но оно не предназначено для получения доступа изолированно с помощью PDP (Пункт выбора стратегии). Таким образом, этот элемент не предназначен для самостоятельного формирования основы решения об авторизации. Он предназначен для изолированного существования только внутри PAP (Пункт управления стратегией) XACML, где он может формировать основной блок управления и многократно использоваться во множестве стратегий.

В элементе `<Policy>` содержится набор элементов `<Rule>` и процедура, установленная для объединения результатов их оценки. Он является основным блоком стратегии, используемой PDP и, таким образом, предназначается для формирования основы решения об авторизации.

В элементе `<PolicySet>` содержится набор `<Policy>` или других элементов `<PolicySet>`, и процедура, установленная для объединения результатов их оценок. Это стандартное средство для объединения отдельных стратегий в единую объединенную стратегию.

7.1.3 Алгоритмы объединения

В XACML определено несколько алгоритмов объединения, которые могут быть идентифицированы с помощью атрибута `RuleCombiningAlgId` или `PolicyCombiningAlgId` элементов `<Policy>` или `<PolicySet>`, соответственно. Алгоритм объединения правил определяет процедуру для принятия решения об авторизации по заданным индивидуальным результатам оценки набора правил. Аналогично, алгоритм объединения стратегий определяет процедуру для принятия решения об авторизации по заданным индивидуальным результатам оценки набора стратегий. Стандартные объединяющие алгоритмы определены для:

- Deny-overrides – Запрет замен (упорядоченных и неупорядоченных);
- Permit-overrides – Разрешение замен(упорядоченных и неупорядоченных);
- First-applicable – Первый-применим; и
- Only-one-applicable – Только один применим.

В случае алгоритма Deny-overrides, если встречается один элемент `<Rule>` или `<Policy>`, оценивающийся, как "Deny", то, независимо от результата оценки других элементов `<Rule>` или `<Policy>` в применяемой стратегии, объединенным результатом будет "Deny".

Так же в случае с алгоритмом Permit-overrides, если встречается один результат "Permit", то объединенным результатом будет "Permit".

В случае с алгоритмом объединения "First-applicable", объединенный результат будет тем же, что и результат оценки первого элемента `<Rule>`, `<Policy>` или `<PolicySet>` в списке правил, цель которых применима к запросу о принятии решения.

Алгоритм, объединяющий стратегии, "Only-one-applicable" применим только к стратегиям. Результат этого алгоритма объединения гарантирует, что одна и только одна стратегия или стратегический набор применим на основании их целей. Если не применяется ни одна стратегия или стратегический набор, то результатом будет "NotApplicable", но если применяется более одной стратегии или стратегического набора, то результатом будет "Indeterminate". Если применяется именно одна стратегия или стратегический набор, то результатом алгоритма объединения будет результат оценки этой одной стратегии или стратегического набора.

Стратегии и стратегические наборы могут принимать значения параметров, которые изменяют поведение алгоритмов объединения. Однако ни один из стандартных алгоритмов объединения не влияет на параметры.

При необходимости пользователи данной Рекомендации могут определять свои собственные алгоритмы объединения.

7.1.4 Многочисленные субъекты

Посредством стратегии контроля доступа часто налагаются требования на действия более, чем одного субъекта. Например, для стратегии, управляющей выполнением дорогостоящей финансовой транзакции, может потребоваться санкция более, чем одного представителя, действующего в разных областях компетенции. Таким образом, в языке XACML признается, что может быть более одного субъекта, имеющего отношение к запросу о принятии решения. Атрибут, называемый "subject-category", используется для различения субъектов, действующих в разных областях компетенции. Установлено несколько стандартных значений для этого атрибута, а пользователи могут определить дополнительные значения.

7.1.5 Стратегии, основанные на атрибутах субъектов и ресурсов

Другим часто встречающимся требованием является учреждение решения об авторизации по некоторым характеристикам субъекта, а не только по его идентичности. Возможно, самым распространенным применением этой идеи является роль субъекта. В XACML предоставляются возможности для поддержания такого подхода. Атрибуты субъекта, содержащиеся в контексте запроса, могут быть идентифицированы с помощью элемента `<SubjectAttributeDesignator>`. В этом элементе содержится URN (Имя унифицированного ресурса), которое идентифицирует этот атрибут. С другой стороны, элемент `<AttributeSelector>` может содержать выражение XPath (Язык ветви XML) по всему контексту запроса для идентификации конкретного значения атрибута субъекта по его местоположению в контексте.

В XACML предоставлен стандартный способ для ссылок на атрибуты, определенные в IETF RFC 2253. Это предназначено для того, чтобы побудить реализаторов использовать стандартные идентификаторы атрибутов для некоторых распространенных атрибутов субъекта.

Другим часто встречающимся требованием является учреждение решения об авторизации по некоторым характеристикам ресурса, а не только по его идентичности. В XACML предоставляются возможности для поддержания такого подхода. Атрибуты ресурса могут быть идентифицированы с помощью элемента `<ResourceAttributeDesignator>`. В этом элементе содержится URN, которое идентифицирует этот атрибут. С другой стороны, элемент `<AttributeSelector>` может содержать выражение XPath по всему контексту запроса для идентификации конкретного значения атрибута ресурса по его местоположению в контексте.

7.1.6 Многозначные атрибуты

Самые распространенные методы для взаимодействующих атрибутов (LDAP, XPath, SAML и т. п.) поддерживают множество значений для каждого атрибута. Таким образом, если PDP XACML занят поиском значения именованного атрибута, то результат может состоять из множества значений. Совокупность таких значений называется "мешок". "Мешок" отличается от набора тем, что в нем могут находиться повторяющиеся значения, в то время как в наборе такого нет. Иногда такая ситуация представляет ошибку. Иногда правило XACML считается исполненным, если какое-нибудь одно из значений атрибута отвечает критерию, выраженному в этом правиле.

В XACML предоставлен набор функций, который позволяет автору стратегии совершенно четко представлять каким образом PDP должен поступать в случае множества значений атрибутов. Это функции "высшего порядка".

7.1.7 Стратегии, основанные на содержании ресурса

Во многих приложениях при принятии решения об авторизации требуется брать за основу данные, содержащиеся в ресурсе, к которому запрашивается допуск. Например, распространенным компонентом стратегии секретности является такой, когда кому-то должно быть дано разрешение на считывание записей, для которых он или она является субъектом. В соответствующей стратегии должна содержаться ссылка на идентифицируемый субъект в самом информационном ресурсе.

В XACML предоставляются возможности для выполнения этого требования, если информационный ресурс может быть представлен, как документ XML. Элемент `<AttributeSelector>` может содержать выражение XPath по всему контексту запроса для идентификации данных в информационном ресурсе, для его использования в оценке стратегии.

7.1.8 Оператор

Стратегии информационной безопасности осуществляют действия над атрибутами субъектов, ресурса, действия и среды с целью принятия решения об авторизации. В процессе принятия решения об авторизации возможно потребуется сравнить или рассчитать атрибуты множества разных типов. Например, в финансовом приложении, возможно потребуется рассчитать доступную сумму кредита для индивидуального лица, прибавляя его/ее кредитный лимит к его/ее остатку на счете. Затем, возможно, потребуется сравнить результат с величиной транзакции. Такого рода ситуации вызывают необходимость арифметических действий над атрибутами субъекта (остаток на счету и кредитный лимит) и ресурсом (величина транзакции).

Еще чаще стратегия может идентифицировать множество ролей, которым разрешено исполнять определенное действие. В соответствующую операцию включена проверка на наличие непустого пересечения между множеством ролей, в которых занят субъект и множеством ролей, идентифицированных этой стратегии. Отсюда необходимость операций над множеством.

В XACML включено несколько встроенных функций и метод добавления нестандартных функций. Эти функции могут быть вложены для произвольного построения сложных выражений. Эта цель достигается с помощью элемента `<Apply>`. У элемента `<Apply>` имеется атрибут XML, называемый `FunctionId`, идентифицирующий функцию, которую следует применить к содержанию элемента. Каждая стандартная функция определена для конкретных комбинаций типов-данных аргумента, а также установлены ее возвращаемые типы-данные. Таким образом, непротиворечивость типов-данных данной стратегии может быть проверена во время написания или анализа стратегии. А типы значений данных, представленных в контексте запроса, могут быть проверены по сравнению с ожидаемыми стратегией значениями, для гарантии предсказуемого результата.

Дополнительно к операторам для численных аргументов и аргументов множеств, определены операторы для аргументов дат, времени и длительности.

Операторы взаимосвязи (равенство и сравнение) также определены для некоторых типов-данных, включая IETF RFC 822 и X.500 имена-формы, строки, идентификаторы URI и т. п.

Также стоит упомянуть операторы логических типов-данных, которые разрешают логические комбинации предикат в правиле. Например, в правиле может содержаться утверждение о том, что доступ может быть разрешен в течение часов работы предприятия AND с оконечных устройств в помещении предприятия.

7.1.9 Распределение стратегии

В распределенной системе отдельные утверждения стратегии могут быть написаны несколькими авторами стратегии и осуществлены в нескольких пунктах осуществления. В дополнение к упрощению сбора и объединения независимых компонентов стратегии, такой подход позволяет модернизировать стратегии соответственно требованиям. Утверждения стратегии XACML могут распределяться любым из нескольких способов. Однако в XACML не дано описание нормативного способа для этой цели. Независимо от средств распределения, ожидается, что пункты PDP подтвердят с помощью проверки элемента стратегии `<Target>`, что данная стратегия применима к запросу о принятии решения, находящемуся в обработке.

Элементы `<Policy>` могут прикрепляться к информационным ресурсам, к которым они применяются. С другой стороны, элементы `<Policy>` могут сохраняться в одном или более мест, где они отыскиваются для проведения оценки. В таких случаях ссылку на применяемую стратегию может давать идентификатор или устройство обнаружения, тесно связанное с информационным ресурсом.

7.1.10 Индексация стратегии

Для эффективности оценки и легкости в управлении вся стратегия безопасности, действующая на предприятии, может быть выражена в виде множества независимых компонентов стратегии. В этом случае необходимо идентифицировать и искать применяемое утверждение стратегии, и проверять правильность утверждения для запрашиваемого действия до его оценки. В этом состоит цель элемента `<Target>` в XACML.

Поддерживаются два подхода:

1) Утверждения стратегии могут храниться в базе данных. В этом случае должен быть сделан запрос с помощью PDP для отыскания именно тех стратегий, которые применимы к набору запросов о принятии решения, на которые ожидается ответ. Дополнительно, PDP должен оценить элемент `<Target>` найденной стратегии или утверждения набора стратегий.

2) С другой стороны, в PDP могут быть загружены все доступные стратегии и проведена оценка их элементов `<Target>` в контексте конкретного запроса о принятии решения, для того чтобы идентифицировать те стратегии или наборы стратегий, которые применимы к данному запросу.

7.1.11 Уровень абстракции

Пункты PEP встречаются во многих формах. Например, PEP может быть частью шлюза удаленного доступа, частью веб-сервера или частью агента-пользователя электронной почты. Нереально ожидать, что все пункты PEP на предприятии в настоящее время или в будущем будут выдавать запросы о принятии решения для PDP в общем формате. Тем не менее, какая-то конкретная стратегия, возможно, должна осуществляться множеством PEP. Неэффективно было бы заставлять автора стратегии писать одну и ту же стратегию несколькими разными способами, для того чтобы они соответствовали требованиям к формату для каждого PEP. Аналогично, атрибуты могут быть заключены в различных типах конвертов (например, сертификаты атрибута X.509, утверждения атрибута SAML и т. п.). Таким образом, существует необходимость в канонической форме запроса и ответа, обрабатываемого пунктом PDP XACML. Такая каноническая форма называется контекстом XACML. Его синтаксис определен в схеме XML.

Конечно, PEP, совместимые с XACML, могут выдавать запросы и получать ответы в форме контекста XACML. Однако, если ситуация другая, то потребуется промежуточный шаг, для того чтобы провести преобразование между форматом запроса/ответа, который воспринимает PEP и форматом контекста XACML, который воспринимает PDP.

Преимуществом такого подхода является то, что стратегии могут быть написаны и проанализированы независимо от конкретной среды, в которой они должны выполняться.

В случае, если собственный формат запрос/ответ установлен в Схеме XML (например, PEP, совместимые с SAML), то преобразование между собственным форматом и контекстом XACML может быть установлено в форме Расширяемого языка таблицы стилей.

Аналогично, в случае, если ресурс, к которому запрашивается доступ, является документом XML, то сам ресурс может быть включен в контекст запроса или на него будет дана ссылка в этом контексте. Затем, посредством использования выражений XPath в стратегии, значения данного ресурса могут быть включены в оценку этой стратегии.

7.1.12 Действия, выполняемые одновременно с исполнением

Во многих приложениях стратегии устанавливают действия, которые должны быть выполнены либо вместо, либо в дополнение к действиям, которые могут выполняться. В XACML предоставлены возможности для установления действий, которые должны выполняться одновременно с оценкой стратегии в элементе `<Obligations>`. Не существует стандартных определений для этих действий в версии 2.0 языка XACML. Таким образом, для правильной интерпретации требуется двустороннее соглашение между PAP и PEP, которое будет исполнять его стратегии. Требуются пункты PEP, которые согласуются с v2.0 языка XACML, для того чтобы отрицать доступ до тех пор, пока они не поймут и не смогут выпустить все элементы `<Obligations>`, связанные с применяемой стратегией. Элементы `<Obligations>` возвращают в PEP для исполнения.

7.2 Модели XACML

Данный пункт является информативным.

Модель потока данных и языковая модель XACML описаны в следующих пунктах.

7.2.1 Модель потока данных

Основные действующие субъекты в домене XACML показаны на диаграмме потока данных рисунка 7-1.

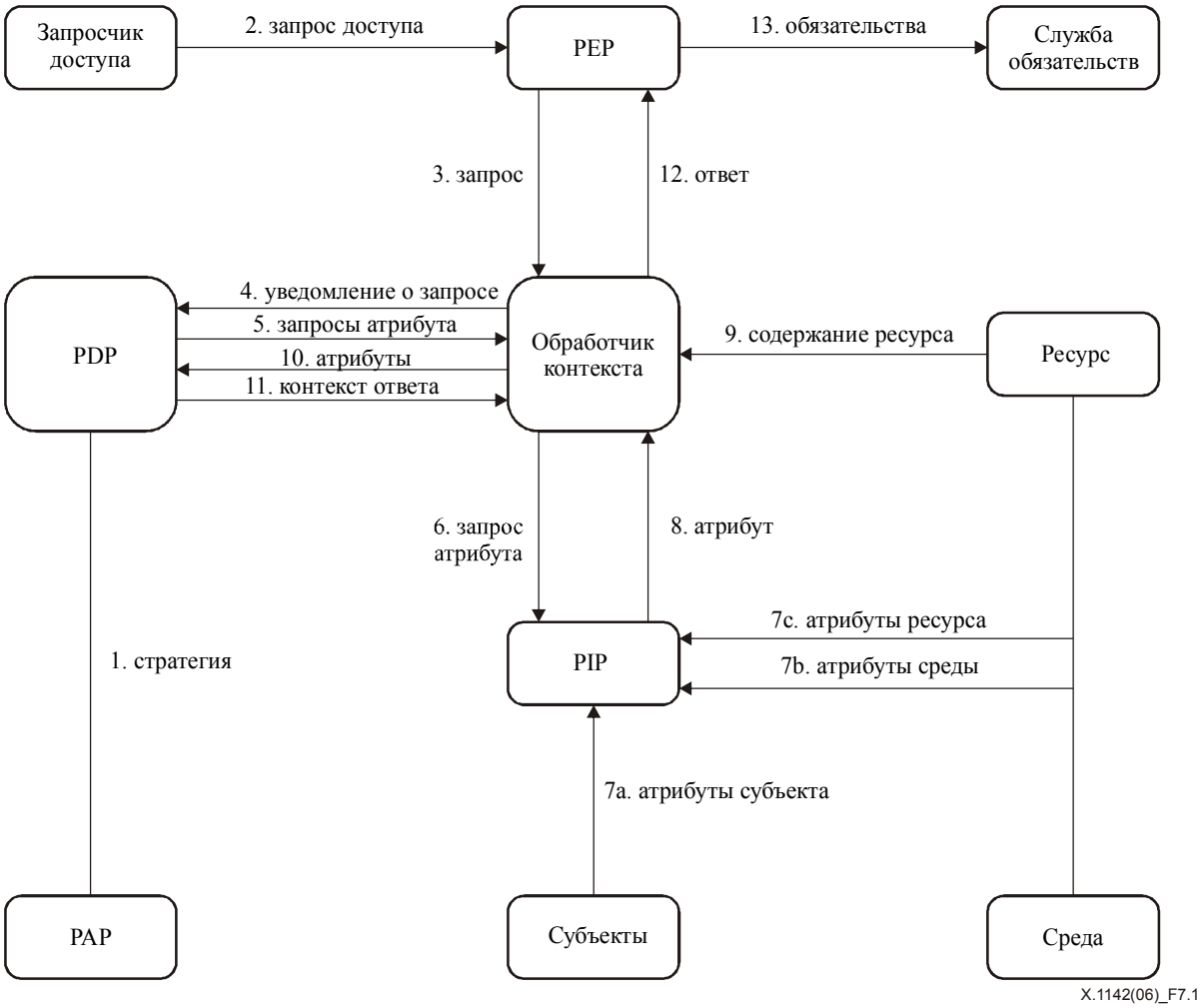

Рисунок 7-1/X.1142 –Диаграмма потока данных

ПРИМЕЧАНИЕ. – Некоторые из потоков данных, показанные на этой диаграмме, могут быть упрощены с помощью репозитария. Например, связи между обработчиком контекста и PIP, или связи между PDP и PAP могут быть упрощены с помощью репозитария. В данной Рекомендации нет намерения устанавливать ограничения на местоположение какого-либо такого репозитария или действительно предписывать конкретный протокол связи для любого из этих потоков данных.

Данная модель действует в следующей последовательности.

1) В пунктах PAP пишутся стратегии или стратегические наборы, и они становятся доступными для PDP. Эти стратегии или стратегические наборы представляют законченную стратегию для установленной цели.
2) Запросчик доступа отправляет запрос о доступе в PEP.

3) PEP отправляет запрос о доступе в обработчик контекста в своем собственном формате запроса, в необязательном порядке включая атрибуты субъектов, ресурса, действия и среды.

4) Обработчик контекста создает контекст запроса XACML и отправляет его в PDP.

5) PDP запрашивает любые дополнительные атрибуты субъекта, ресурса, действия и среды у обработчика контекста.

6) Обработчик контекста запрашивает атрибуты у PIP.

7) PIP получает запрошенные атрибуты.

8) PIP возвращает запрошенные атрибуты в обработчик контекста.

9) Необязательно, обработчик контекста включает ресурс в контекст.

10) Обработчик контекста отправляет запрошенные атрибуты и (необязательно) ресурс в PDP. PDP оценивает стратегию.

11) PDP возвращает контекст ответа (включая решение об авторизации) в обработчик контекста.

12) Обработчик контекста переводит контекст ответа в собственный формат ответа PEP. Обработчик контекста возвращает ответ в PEP.

13) PEP выполняет обязательства.

14) (Не показано) Если доступ разрешен, тогда PEP разрешает доступ к ресурсу; иначе, он запрещает доступ.

7.3 Контекст XACML

Язык XACML должен подходить для многообразных прикладных сред. Основной язык изолирован от прикладной среды контекстом XACML, как показано на рисунке 7-2, на котором область применения XACML указана затененной областью. Контекст XACML определен в схеме XML, описывающей каноническое представление для входов и выходов PDP. Атрибуты, на которые ссылаются с помощью экземпляра стратегии XACML, могут быть в форме выражений XPath по контексту или указателей атрибутов, которые идентифицируют атрибут с помощью субъекта, ресурса, действия или окружающей среды и его идентификатор, тип данных и (необязательно) его запрашивающая сторона. Реализации должны производить преобразования между представлениями атрибута в прикладной среде (например, SAML) и представлениями атрибута в контексте XACML. Способы достижения такого результата выходят за рамки данной Рекомендации. В некоторых случаях, таких как SAML, это преобразование может быть завершено автоматически посредством использования трансформации XSLT (см. W3C XSLT:1999).

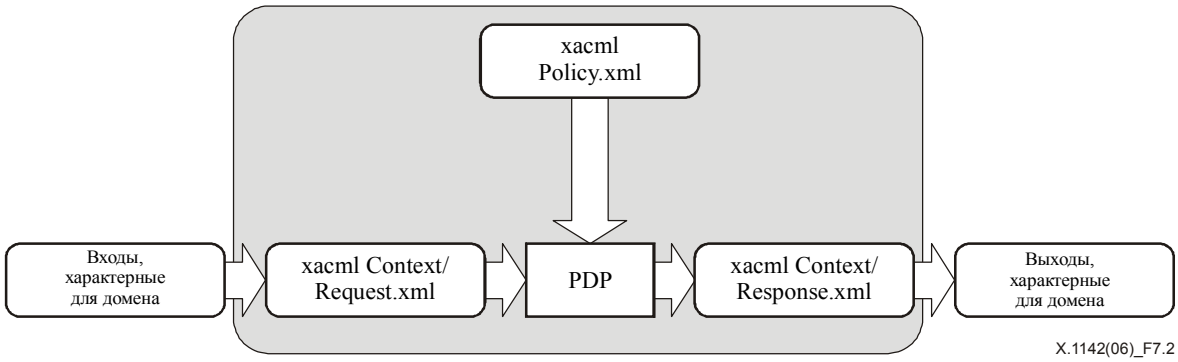

Рисунок 7-2/X.1142 – Контекст XACML

Не требуется, чтобы PDP работал напрямую с представлением стратегии XACML. Он может работать напрямую с альтернативным представлением.

7.3.1 Модель языка стратегии

Модель языка стратегии показана на рисунке 7-3. Основными компонентами этой модели являются:

- Правило;
- Стратегия; и
- Стратегический набор.

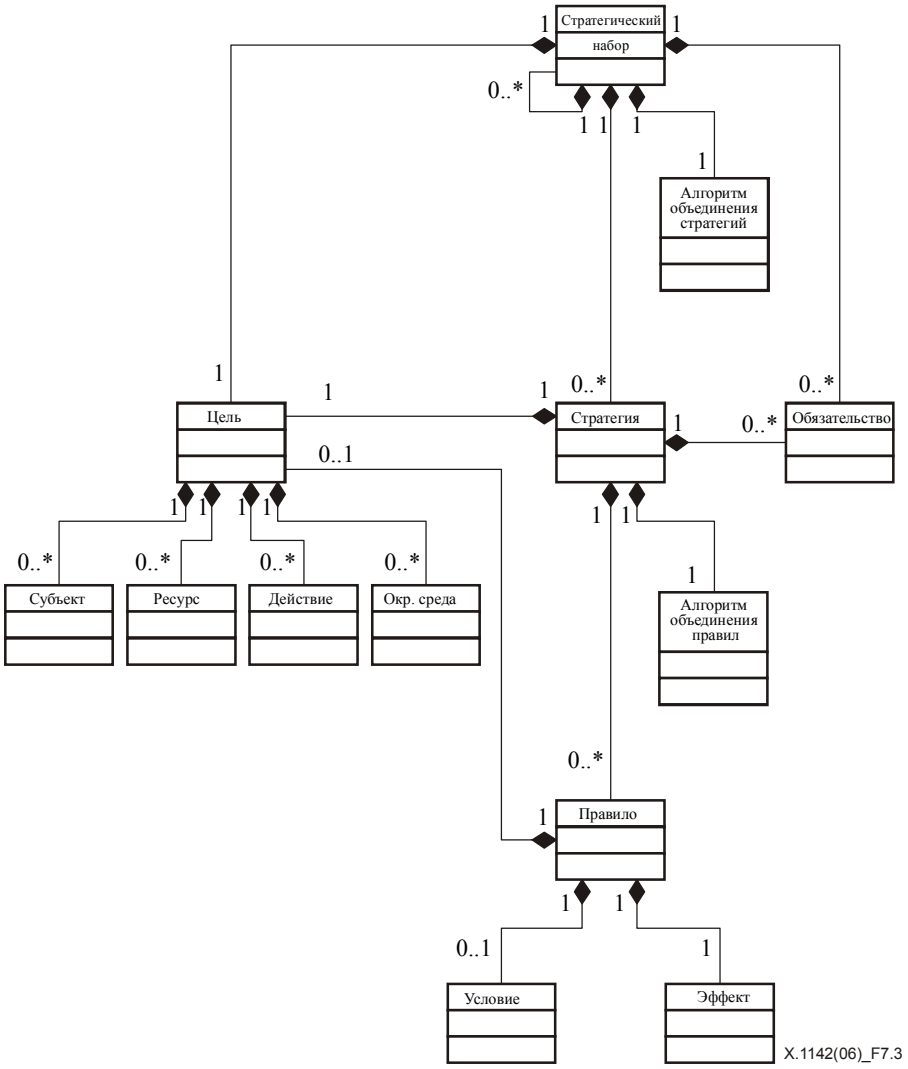

Рисунок 7-3/X.1142 – Модель языка стратегии

7.3.1.1 Правило

Правило – это самая элементарная единица стратегии. Оно может существовать изолированно только внутри одного из основных действующих субъектов домена XACML. Для обмена правилами между основными действующими субъектами, они могут инкапсулироваться в стратегию. Правило может оцениваться на основе его содержания. Основными компонентами правила являются:

- цель;
- эффект; и
- условие.

7.3.1.1.1 Цель правила

Целью определяется набор из:

- ресурсов;
- субъектов;
- действий; и
- среда,

для применения к которым это правило предназначено. Элемент `<Condition>` может еще больше уточнить применимость, установленную этой целью. Если данное правило предназначено для применения ко всем объектам конкретного типа-данных, тогда соответствующий объект не включается в цель. PDP XACML проверяет, чтобы пары, определенные этой целью, выполнялись с помощью атрибутов субъектов, ресурса, действия и окружающей среды в контексте запроса. Определения цели являются обособленными, для того чтобы подходящие правила могли быть эффективно идентифицированы с помощью PDP.

Элемент <Target> может отсутствовать в <Rule>. В этом случае, цель <Rule> будет той же самой, что и у родительского элемента <Policy>.

Некоторые имена-формы субъекта, имена-формы ресурса и некоторые типы ресурса являются внутренне структурированными. Например, имя-форма справочника X.500 и имя-форма IETF RFC 822 являются структурированными именами-формами субъекта, в то время, как у учетного номера обычно явной структуры нет. Путевые имена системных файлов UNIX и идентификаторы URI являются примерами структурированных имен-форм ресурса. А документ XML является примером структурированного ресурса.

В общем случае, имя узла (кроме краевого узла) в структурированной имя-форме является также законным экземпляром имя-формы. Итак, например, имя "med.example.com" IETF RFC 822 является законным именем IETF RFC 822, идентифицирующим набор почтовых адресов, которые принимаются почтовым сервером med.example.com. А значение XPath/XPointer //xacml-context:Request/xacml-context:Resource/xacml-context:ResourceContent/md:record/md:patient/ является легальным значением XPath/XPointer, идентифицирующим набор узлов в документе XML.

Возникает вопрос: как должно интерпретироваться имя, которое идентифицирует набор субъектов или ресурсов с помощью PDP, если оно появляется в стратегии или в контексте запроса? Предназначены ли они только для представления узла, явно идентифицированного по имени, или они предназначены для представления всего поддерева, которое подчинено этому узлу?

В случае субъектов не существует реального объекта, который бы относился к такому узлу. Итак, имена такого типа всегда относятся к набору субъектов, подчиненных в структуре имен идентифицированному узлу. Следовательно, не должны использоваться имена некраевых субъектов в функциях равенства, только в функциях соответствий, таких как "urn:oasis:names:tc:xacml:1.0:function:rfc822Name-match", а не "urn:oasis:names:tc:xacml:1.0:function:rfc822Name-equal".

7.3.1.1.2 Эффект

Эффект этого правила показывает запланированную автором правила последовательность оценок "True" для этого правила. Разрешены два значения: "Permit" и "Deny".

7.3.1.1.3 Условие

Условие представляет логическое выражение, которое уточняет применимость этого правила за пределами предикат, подразумеваемых его целью. Таким образом, оно может отсутствовать.

7.3.1.2 Стратегия

Из модели потока данных можно увидеть, что системные объекты не обмениваются правилами. Таким образом, PAP объединяет правила в стратегии. Стратегия содержит четыре основных компонента:

- цель;
- идентификатор алгоритма объединения правил;
- набор правила; и
- обязательства.

7.3.1.2.1 Цель стратегии

Элемент <PolicySet>, <Policy> или <Rule> XACML включает в себя элемент <Target>, который устанавливает набор субъектов, ресурсов, действий и сред, к которым он применяется. Элемент <Target> элемента <PolicySet> или <Policy> может быть заявлен автором элемента <PolicySet> или <Policy>, или он может быть рассчитан из элементов <Target> элементов <PolicySet>, <Policy> и <Rule>, которые он содержит.

Системный объект, который рассчитывает <Target> таким образом, не определяется с помощью XACML, но существует два логических метода, которыми можно воспользоваться. В одном методе, элемент <Target> внешних <PolicySet> или <Policy> ("внешний компонент") рассчитывается, как объединение всех элементов <Target> упомянутых элементов <PolicySet>, <Policy> или <Rule> ("внутренних компонентов"). В другом методе, элемент <Target> внешнего компонента рассчитывается, как пересечение всех элементов <Target> внешних компонентов. Результаты оценки в каждом случае будут очень разными: в первом случае элемент <Target> внешнего компонента делает его применимым к любому запросу о принятии решения, которое подходит элементу <Target>, по крайней мере, одного внутреннего компонента; во втором случае, элемент <Target> внешнего компонента делает его применимым только к тем запросам о принятии решения, которые подходят элементам <Target> каждого внутреннего компонента. Заметим, что вычисление пересечения набора элементов <Target>, вероятно, только тогда имеет практический смысл, когда модель-данные цели сравнительно простые.

В случаях, если <Target> элемента <Policy> заявлен автором стратегии, то любые элементы <Rule> компонента в <Policy>, у которых те же самые элементы <Target>, что и у элемента <Policy>, могут не включать элемент <Target>. Такие элементы <Rule> наследуют <Target> элемента <Policy>, в которых они содержатся.

7.3.1.2.2 Алгоритм объединения правил

Алгоритм объединения правил устанавливает процедуру, с помощью которой результаты оценки правил компонентов объединяются при оценке стратегии, например, значение `Decision`, помещенное в контекст ответа с помощью PDP, является значением стратегии, как определено алгоритмом объединения правил. У стратегий могут быть параметры объединения, которые влияют на работу алгоритма объединения правил.

7.3.1.2.3 Обязательства

Обязательства могут добавляться автором стратегии.

Если PDP оценивает стратегию, содержащую обязательства, то он возвращает некоторые из тех обязательств в PEP в контексте ответа.

7.3.1.3 Стратегический набор

Стратегический набор содержит четыре основных компонента:

- цель;
- идентификатор алгоритма объединения стратегий;
- стратегический набор; и
- обязательства.

7.3.1.3.1 Алгоритм объединения стратегий

Алгоритм объединения стратегий устанавливает процедуру, с помощью которой результаты оценки стратегий компонентов объединяются при оценке стратегического набора, например, значение `Decision`, помещенное в контекст ответа с помощью PDP, является результатом оценки стратегического набора, как определено алгоритмом объединения стратегий. У стратегического набора могут быть параметры объединения, которые влияют на работу алгоритма объединения стратегий.

7.3.1.3.2 Обязательства

Автор стратегического набора может добавлять обязательства к стратегическому набору в дополнение к тем, которые содержатся в стратегиях компонентов и стратегических наборах.

Если PDP оценивает стратегический набор, содержащий обязательства, то он возвращает некоторые из этих обязательств в PEP в контексте ответа.

7.4 Синтаксис стратегии

Фрагменты схемы в следующих пунктах являются ненормативными.

7.4.1 Элемент <PolicySet>

Элемент `<PolicySet>` является элементом высшего уровня в схеме стратегии XACML. `<PolicySet>` является агрегацией других стратегических наборов и стратегий. Стратегические наборы могут включаться в охватывающий элемент `<PolicySet>` либо напрямую, используя элемент `<PolicySet>`, либо косвенным способом, используя элемент `<PolicySetIdReference>`. Стратегии могут включаться в охватывающий элемент `<PolicySet>` либо напрямую, используя элемент `<Policy>`, либо косвенным способом, используя элемент `<PolicyIdReference>`.

Элемент `<PolicySet>` можно оценивать, в этом случае должна использоваться процедура оценки, определенная в данной Рекомендации.

Если в элементе `<PolicySet>` содержатся ссылки на другие стратегические наборы или стратегии в виде указателей URL, то эти ссылки могут быть отображены.

Стратегические наборы и стратегии, включенные в элемент `<PolicySet>`, должны объединяться с использованием алгоритма, идентифицированного атрибутом `PolicyCombiningAlgId`. С `<PolicySet>` обращаются точно так же, как и с `<Policy>` во всех алгоритмах объединения стратегий.

Элемент `<Target>` определяет применяемость элемента `<PolicySet>` к запросам о принятии решения. Если элемент `<Target>` внутри элемента `<PolicySet>` подходит к контексту запроса, то элемент `<PolicySet>` может использоваться пунктом PDP при принятии решения об авторизации.

Элемент `<Obligations>` содержит набор обязательств, которые должны быть выполнены с помощью PEP вместе с решением об авторизации. Если PEP не воспринимает или не может выполнить какие-либо из этих обязательств, то он должен действовать таким образом, как если бы PDP вернул значение "Deny" в решении об авторизации.

```
<xs:element name="PolicySet" type="xacml:PolicySetType"/>
<xs:complexType name="PolicySetType">
   <xs:sequence>
         <xs:element ref="xacml:Description" minOccurs="0"/>
         <xs:element ref="xacml:PolicySetDefaults" minOccurs="0"/>
         <xs:element ref="xacml:Target"/>
         <xs:choice minOccurs="0" maxOccurs="unbounded">
               <xs:element ref="xacml:PolicySet"/>
               <xs:element ref="xacml:Policy"/>
               <xs:element ref="xacml:PolicySetIdReference"/>
               <xs:element ref="xacml:PolicyIdReference"/>
               <xs:element ref="xacml:CombinerParameters"/>
               <xs:element ref="xacml:PolicyCombinerParameters"/>
               <xs:element ref="xacml:PolicySetCombinerParameters"/>
         </xs:choice>
         <xs:element ref="xacml:Obligations" minOccurs="0"/>
   </xs:sequence>
   <xs:attribute name="PolicySetId" type="xs:anyURI" use="required"/>
   <xs:attribute name="Version" type="xacml:VersionType" default="1.0"/>
   <xs:attribute name="PolicyCombiningAlgId" type="xs:anyURI" use="required"/>
</xs:complexType>
```

Элемент <PolicySet> является элементом составного типа **PolicySetType**.

Элемент <PolicySet> содержит следующие атрибуты и элементы:

- PolicySetId [Обязательный]

  Идентификатор стратегического набора. В обязанности PAP входит гарантирование того, что никакие две стратегии, находящиеся в поле зрения PDP, не имеют один и тот же идентификатор. Этой цели можно добиться, следуя предписанной схеме URN или URI. Если идентификатор стратегического набора в виде URL, то это может быть отображено.

- Version [По умолчанию 1.0]

  Номер версии PolicySet.

- PolicyCombiningAlgId [Обязательный]

  Идентификатор алгоритма объединения стратегий, с помощью которого должны объединяться компоненты <PolicySet>, <CombinerParameters>, <PolicyCombinerParameters> и <PolicySetCombinerParameters>.

- <Description> [Необязательный]

  Описание стратегического набора в произвольной форме.

- <PolicySetDefaults> [Необязательный]

  Набор значений по умолчанию, применимый к этому стратегическому набору. Областью применения элемента <PolicySetDefaults> должен быть прилагающийся стратегический набор.

- <Target> [Обязательный]

  Элемент <Target> определяет применимость стратегического набора к набору запросов о принятии решения.

  Элемент <Target> может быть заявлен создателем <PolicySet> или он может быть вычислен из элементов <Target> упомянутых элементов <Policy>, либо методом дизъюнкции, либо методом объединения.

- <PolicySet> [Любое количество]

  Стратегический набор, который включен в данный стратегический набор.

- <Policy> [Любое количество]

  Стратегия, которая включена в данный стратегический набор.

- <PolicySetIdReference> [Любое количество]

  Ссылка на стратегический набор, который должен быть включен в данный стратегический набор. Если <PolicySetIdReference> является URL, то это может быть отображено.

– <PolicyIdReference> [Любое количество]

Ссылка на стратегию, которая должна быть включена в данный стратегический набор. Если <PolicyIdReference> является URL, то это может быть отображено.

– <Obligations> [Необязательный]

Содержит набор элементов <Obligation>.

– <CombinerParameters> [Необязательный]

Содержит последовательность элементов <CombinerParameter>.

– <PolicyCombinerParameters> [Необязательный]

Содержит последовательность элементов <CombinerParameter>, которые связаны с конкретным элементом <Policy> или <PolicyIdReference> внутри <PolicySet>.

– <PolicySetCombinerParameters> [Необязательный]

Содержит последовательность элементов <CombinerParameter>, которые связаны с определенным элементом <PolicySet> или <PolicySetIdReference> внутри <PolicySet>.

7.4.2 Элемент <Description>

Элемент <Description> содержит описание элемента <PolicySet>, <Policy> или <Rule> в произвольной форме. Элемент <Description> является элементом простого типа **xs:string**.

```
<xs:element name="Description" type="xs:string"/>
```

7.4.3 Элемент <PolicySetDefaults>

Элемент <PolicySetDefaults> должен устанавливать значения по умолчанию, которые применимы к элементу <PolicySet>.

```
<xs:element name="PolicySetDefaults" type="xacml:DefaultsType"/>
<xs:complexType name="DefaultsType">
   <xs:sequence>
         <xs:choice>
               <xs:element ref="xacml:XPathVersion" minOccurs="0"/>
         </xs:choice>
   </xs:sequence>
</xs:complexType>
```

Элемент <PolicySetDefaults> является элементом составного типа **DefaultsType**.

Элемент <PolicySetDefaults> содержит следующие элементы:

– <XPathVersion> [Необязательный]

Версия по умолчанию XPath.

7.4.4 Элемент <XPathVersion>

Элемент <XPathVersion> должен устанавливать версию W3C XPath:1999, которая должна использоваться элементами <AttributeSelector> функциями на основе XPath в данном стратегическом наборе или стратегии.

```
<xs:element name="XPathVersion" type="xs:anyURI"/>
```

Идентификатором URI для W3C XPath:1999 является "http://www.w3.org/TR/1999/Rec-xpath-19991116". Требуется элемент <XPathVersion>, если в прилагаемом стратегическом наборе или стратегии XACML содержатся элементы <AttributeSelector> или функции на основе XPath.

7.4.5 Элемент <Target>

Элемент <Target> идентифицирует набор запросов о принятии решения, который родительский элемент собирается оценить. Элемент <Target> должен появиться, как дочерний элемент <PolicySet> и <Policy> и может появиться, как дочерний элемент <Rule>. В нем содержатся определения для субъектов, ресурсов, действий и сред.

Элемент <Target> должен содержать конъюнктивную последовательность элементов <Subjects>, <Resources> <Actions> и <Environments>. Для родителя элемента <Target>, чтобы он был применим к запросу о принятии решения, должно существовать, по крайней мере, одно положительное сопоставление между каждой секцией элемента <Target> и соответствующей секцией элемента <xacml-context:Request>.

```
<xs:element name="Target" type="xacml:TargetType"/>
<xs:complexType name="TargetType">
   <xs:sequence>
         <xs:element ref="xacml:Subjects" minOccurs="0"/>
         <xs:element ref="xacml:Resources" minOccurs="0"/>
         <xs:element ref="xacml:Actions" minOccurs="0"/>
         <xs:element ref="xacml:Environments" minOccurs="0"/>
   </xs:sequence>
</xs:complexType>
```

Элемент <Target> является элементом составного типа **TargetType**.

В элементе <Target> содержатся следующие элементы:

– <Subjects> [Необязательный]

  Спецификация сопоставления для атрибутов субъектов в контексте. Если этот элемент пропущен, то цель должна подходить всем субъектам.

– <Resources> [Необязательный]

  Спецификация сопоставления для атрибутов ресурсов в контексте. Если этот элемент пропущен, то цель должна подходить всем ресурсам.

– <Actions> [Необязательный]

  Спецификация сопоставления для атрибутов действий в контексте. Если этот элемент пропущен, то цель должна подходить всем действиям.

– <Environments> [Необязательный]

  Спецификация сопоставления для атрибутов сред в контексте. Если этот элемент пропущен, то цель должна подходить всем средам.

7.4.6 Элемент <Subjects>

В элементе <Subjects> должна содержаться дизъюнктивная последовательность элементов <Subject>.

```
<xs:element name="Subjects" type="xacml:SubjectsType"/>
<xs:complexType name="SubjectsType">
   <xs:sequence>
         <xs:element ref="xacml:Subject" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

Элемент <Subjects> является элементом составного типа **SubjectsType**.

В элементе <Subjects> содержатся следующие элементы:

– <Subject> [От одного до многих, Обязательный]

  Как определено в пункте 7.4.7.

7.4.7 Элемент <Subject>

В элементе <Subject> должна содержаться конъюнктивная последовательность элементов <SubjectMatch>.

```
<xs:element name="Subject" type="xacml:SubjectType"/>
<xs:complexType name="SubjectType">
   <xs:sequence>
         <xs:element ref="xacml:SubjectMatch" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

Элемент <Subjects> является элементом составного типа **SubjectType**.

В элементе <Subjects> содержатся следующие элементы:

– `<SubjectMatch>` [От одного до многих]

  Конъюнктивная последовательность индивидуальных сопоставлений атрибутов субъекта в контексте запроса и встроенные значения атрибутов.

7.4.8 Элемент <SubjectMatch>

Элемент `<SubjectMatch>` должен идентифицировать набор объектов, связанных с субъектом, методом сопоставлений значений атрибута в элементе `<xacml-context:Subject>` контекста запроса с встроенным значением атрибута.

```
<xs:element name="SubjectMatch" type="xacml:SubjectMatchType"/>
<xs:complexType name="SubjectMatchType">
   <xs:sequence>
         <xs:element ref="xacml:AttributeValue"/>
         <xs:choice>
               <xs:element ref="xacml:SubjectAttributeDesignator"/>
               <xs:element ref="xacml:AttributeSelector"/>
         </xs:choice>
   </xs:sequence>
   <xs:attribute name="MatchId" type="xs:anyURI" use="required"/>
</xs:complexType>
```

Элемент `<SubjectMatch>` является элементом составного типа **SubjectMatchType**.

Элемент `<SubjectMatch>` содержит следующие атрибуты и элементы:

– `MatchId` [Обязательный]

  Устанавливает функцию сопоставлений. Значение этого атрибута должно быть типа **xs:anyURI** с легальными значениями такими, как записано в пункте 7.6.5.

– `<xacml:AttributeValue>` [Обязательный]

  Встроенное значение атрибута.

– `<SubjectAttributeDesignator>` [Обязательный выбор]

  может использоваться для идентификации одного или более значений атрибута в элементе `<Subject>` контекста запроса.

– `<AttributeSelector>` [Обязательный выбор]

  может использоваться для идентификации одного или более значений атрибута в контексте запроса. Выражение XPath должно быть принято для атрибута в элементе `<Subject>` контекста запроса.

7.4.9 Элемент <Resources>

В элементе `<Resources>` element должна содержаться дизъюнктивная последовательность элементов `<Resource>`.

```
<xs:element name="Resources" type="xacml:ResourcesType"/>
<xs:complexType name="ResourcesType">
   <xs:sequence>
         <xs:element ref="xacml:Resource" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

Элемент `<Resources>` является элементом составного типа **ResourcesType**.

Элемент `<Resources>` содержит следующие элементы:

– `<Resource>` [От одного до многих, Обязательный]

  Смотрите пункт 7.4.10.

7.4.10 Элемент <Resource>

В элементе `<Resource>` должна содержаться конъюнктивная последовательность элементов `<ResourceMatch>`.

```
<xs:element name="Resource" type="xacml:ResourceType"/>
<xs:complexType name="ResourceType">
   <xs:sequence>
         <xs:element ref="xacml:ResourceMatch" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

Элемент <Resource> является элементом составного типа **ResourceType**.

В элементе <Resource> содержатся следующие элементы:

– <ResourceMatch> [От одного до многих]

Конъюнктивная последовательность индивидуальных сопоставлений атрибутов ресурса в контексте запроса и встроенных значений атрибутов.

7.4.11 Элемент <ResourceMatch>

Элемент <ResourceMatch> должен идентифицировать набор объектов, связанных с ресурсом, сопоставлением значений атрибута в элементе <xacml-context:Resource> контекста запроса с встроенным значением атрибута.

```
<xs:element name="ResourceMatch" type="xacml:ResourceMatchType"/>
<xs:complexType name="ResourceMatchType">
   <xs:sequence>
         <xs:element ref="xacml:AttributeValue"/>
         <xs:choice>
               <xs:element ref="xacml:ResourceAttributeDesignator"/>
               <xs:element ref="xacml:AttributeSelector"/>
         </xs:choice>
   </xs:sequence>
   <xs:attribute name="MatchId" type="xs:anyURI" use="required"/>
</xs:complexType>
```

Элемент <ResourceMatch> является элементом составного типа **ResourceMatchType**.

В элементе <ResourceMatch> содержатся следующие атрибуты и элементы:

– MatchId [Обязательный]

Устанавливает функцию сопоставлений. Значения этого атрибута должны быть типа **xs:anyURI**, с легальными значениями такими, как записано в пункте 7.6.5.

– <xacml:AttributeValue> [Обязательный]

Встроенное значение атрибута.

– <ResourceAttributeDesignator> [Обязательный выбор]

может использоваться для идентификации одного или более значений атрибута в элементе <Resource> контекста запроса.

– <AttributeSelector> [Обязательный выбор]

может использоваться для идентификации одного или более значений атрибута в контексте запроса. Выражение XPath должно быть принято для атрибута в элементе <Resource> контекста запроса.

7.4.12 Элемент <Actions>

В элементе <Actions> должна содержаться дизъюнктивная последовательность элементов <Action>.

```
<xs:element name="Actions" type="xacml:ActionsType"/>
<xs:complexType name="ActionsType">
   <xs:sequence>
         <xs:element ref="xacml:Action" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

Элемент <Actions> является элементом составного типа **ActionsType**.

В элементе <Actions> содержатся следующие элементы:

– <Action> [От одного до многих, Обязательный]

Смотрите пункт 7.4.13.

7.4.13 Элемент <Action>

В элементе <Action> должна содержаться конъюнктивная последовательность элементов <ActionMatch>.

```
<xs:element name="Action" type="xacml:ActionType"/>
<xs:complexType name="ActionType">
```

```
   <xs:sequence>
          <xs:element ref="xacml:ActionMatch" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

Элемент <Action> является составным элементом типа **ActionType**.

В элементе <Action> содержатся следующие элементы:

– <ActionMatch> [От одного до многих]

  Конъюнктивная последовательность индивидуальных сопоставлений атрибутов действия в контексте запроса и встроенных значений атрибутов, смотрите пункт 7.4.14.

7.4.14 Элемент <ActionMatch>

Элемент <ActionMatch> должен идентифицировать набор объектов, связанных с действием, сопоставлением значений атрибута в элементе <xacml-context:Action> контекста запроса с встроенным значением атрибута.

```
<xs:element name="ActionMatch" type="xacml:ActionMatchType"/>
<xs:complexType name="ActionMatchType">
   <xs:sequence>
          <xs:element ref="xacml:AttributeValue"/>
          <xs:choice>
                 <xs:element ref="xacml:ActionAttributeDesignator"/>
                 <xs:element ref="xacml:AttributeSelector"/>
          </xs:choice>
   </xs:sequence>
   <xs:attribute name="MatchId" type="xs:anyURI" use="required"/>
</xs:complexType>
```

Элемент <ActionMatch> является элементом составного типа **ActionMatchType**.

В элементе <ActionMatch> содержатся следующие атрибуты и элементы:

– MatchId [Обязательный]

  Устанавливает функцию сопоставлений. Значения этого атрибута должны быть типа **xs:anyURI**, с легальными значениями такими, как записано в пункте 7.6.5.

– <xacml:AttributeValue> [Обязательный]

  Встроенное значение атрибута.

– <ActionAttributeDesignator> [Обязательный выбор]

  может использоваться для идентификации одного или более значений атрибута в элементе <Action> контекста запроса.

– <AttributeSelector> [Обязательный выбор]

  может использоваться для идентификации одного или более значений атрибута в контексте запроса. Выражение XPath должно быть принято для атрибута в элементе <Action> контекста запроса.

7.4.15 Element <Environments>

Элемент <Environments> должен содержать дизъюнктивную последовательность элементов <Environments>.

```
<xs:element name="Environments" type="xacml:EnvironmentsType"/>
<xs:complexType name="EnvironmentsType">
   <xs:sequence>
          <xs:element ref="xacml:Environment" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

Элемент <Environments> является элементом составного типа **EnvironmentsType**.

В элементе<Environments> содержатся следующие элементы:

– <Environment> [От одного до многих, Обязательный]

  Смотрите пункт 7.4.16.

7.4.16 Элемент <Environment>

В элементе <Environment> должна содержаться конъюнктивная последовательность элементов <EnvironmentMatch>.

```
<xs:element name="Environment" type="xacml:EnvironmentType"/>
<xs:complexType name="EnvironmentType">
   <xs:sequence>
          <xs:element ref="xacml:EnvironmentMatch" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

Элемент <Environment> является элементом составного типа **EnvironmentType**.

В элементе<Environment> содержатся следующие элементы:

– <EnvironmentMatch> [От одного до многих]

  Конъюнктивная последовательность индивидуальных сопоставлений атрибутов среды в контексте запроса и встроенных значений атрибутов.

7.4.17 Элемент <EnvironmentMatch>

Элемент <EnvironmentMatch> должен идентифицировать набор объектов, связанных со средой, сопоставлением значений атрибута в элементе <xacml-context:Environment> контекста запроса с встроенным значением атрибута.

```
<xs:element name="EnvironmentMatch" type="xacml:EnvironmentMatchType"/>
<xs:complexType name="EnvironmentMatchType">
   <xs:sequence>
          <xs:element ref="xacml:AttributeValue"/>
          <xs:choice>
                 <xs:element ref="xacml:EnvironmentAttributeDesignator"/>
                 <xs:element ref="xacml:AttributeSelector"/>
          </xs:choice>
   </xs:sequence>
   <xs:attribute name="MatchId" type="xs:anyURI" use="required"/>
</xs:complexType>
```

Элемент <EnvironmentMatch> является элементом составного типа **EnvironmentMatchType**.

В элементе <EnvironmentMatch> содержатся следующие атрибуты и элементы:

– MatchId [Обязательный]

  Устанавливает функцию сопоставлений. Значения этого атрибута должны быть типа **xs:anyURI**, с легальными значениями такими, как записано в пункте 7.6.5.

– <xacml:AttributeValue> [Обязательный]

  Встроенное значение атрибута.

– <EnvironmentAttributeDesignator> [Обязательный выбор]

  может использоваться для идентификации одного или более значений атрибута в элементе <Environment> контекста запроса.

– <AttributeSelector> [Обязательный выбор]

  может использоваться для идентификации одного или более значений атрибута в контексте запроса. Выражение XPath должно быть принято для атрибута в элементе <Environment> контекста запроса.

7.4.18 Элемент <PolicySetIdReference>

Элемент <PolicySetIdReference> должен использоваться для ссылки на элемент <PolicySet> с помощью id. Если <PolicySetIdReference> является указателем URL, то он может быть отображен в элемент <PolicySet>. Однако механизм для отображения ссылки стратегического набора в соответствующий стратегический набор выходит за рамки данной Рекомендации.

```
<xs:element name="PolicySetIdReference" type="xacml:IdReferenceType"/>
xs:complexType name="IdReferenceType">
   <xs:simpleContent>
          <xs:extension base="xs:anyURI">
                 <xs:attribute name="xacml:Version" type="xacml:VersionMatchType"
                 use="optional"/>
```

```
                <xs:attribute name="xacml:EarliestVersion"
type="xacml:VersionMatchType" use="optional"/>
                <xs:attribute name="xacml:LatestVersion"
type="xacml:VersionMatchType" use="optional"/>
          </xs:extension>
   </xs:simpleContent>
</xs:complexType>
```

Элемент <PolicySetIdReference> является элементом составного типа **xacml:IdReferenceType**.

IdReferenceType расширяет тип **xs:anyURI** со следующими атрибутами:

– Version [Необязательный]

Устанавливает выражение сопоставления для версии упомянутого стратегического набора.

– EarliestVersion [Необязательный]

Устанавливает выражение сопоставления для самой ранней допустимой версии упомянутого стратегического набора.

– LatestVersion [Необязательный]

Устанавливает выражение сопоставления для самой поздней допустимой версии упомянутого стратегического набора.

Любая комбинация этих атрибутов может присутствовать в <PolicySetIdReference>. Упомянутый стратегический набор должен сопоставлять все выражения. Если ни одного из этих атрибутов нет, то допустима любая версия стратегического набора. В случае, если получено более одной подходящей версии, надо использовать самую позднюю.

7.4.19 Элемент <PolicyIdReference>

Элемент <xacml:PolicyIdReference> должен использоваться для ссылки на элемент <Policy> с помощью id. Если <PolicyIdReference> является указателем URL, то он может быть отображен в элемент <Policy>. Однако механизм для отображения ссылки стратегии в соответствующую стратегию выходит за рамки данной Рекомендации.

```
<xs:element name="PolicyIdReference" type="xacml:IdReferenceType"/>
```

Элемент <PolicyIdReference> является элементом составного типа **xacml:IdReferenceType**.

7.4.20 VersionType простого типа

В элементах этого типа должен содержаться номер версии стратегии или стратегического набора.

```
<xs:simpleType name="VersionType">
   <xs:restriction base="xs:string">
          <xs:pattern value="(\d+\.)*\d+"/>
   </xs:restriction>
</xs:simpleType>
```

Номер версии выражен в виде десятичных чисел, каждое из которых отделенное периодом (.). 'd+' представляет последовательность одной или более десятичных цифр.

7.4.21 VersionMatchType простого типа

В элементах этого типа содержится ограниченное регулярное выражение, подходящее для номера версии. Это выражение должно подходить для версий упомянутой стратегии или стратегического набора, которые допустимы для их включения в обращение к стратегии или стратегическому набору.

```
<xs:simpleType name="VersionMatchType">
   <xs:restriction base="xs:string">
          <xs:pattern value="((\d+|\*)\.)*(\d+|\*|\+)"/>
   </xs:restriction>
</xs:simpleType>
```

Сопоставление версии является '.'-обособленным, так же как и строка версий. Число представляет прямое численное сопоставление. Символ '*' означает, что любое отдельное число является действительным. Символ '+' означает, что любое число и любые последующие числа являются действительными. Таким образом, все следующие четыре образца подходят строке версий '1.2.3': '1.2.3', '1.*.3', '1.2.*' и '1.+'.

7.4.22 Элемент <Policy>

Элемент <Policy> является самым маленьким объектом, который должен быть представлен в PDP для оценки.

Элемент <Policy> может оцениваться, и в этом случае должна использоваться процедура оценки, определенная в пункте 7.6.10.

Основными компонентами этого элемента являются элементы <Target>, <Rule>, <CombinerParameters>, <RuleCombinerParameters> и <Obligations> и атрибут RuleCombiningAlgId.

Элемент <Target> определяет применимость элемента <Policy> к набору запросов о принятии решения. Если элемент <Target> внутри элемента <Policy> подходит для контекста запроса, то элемент <Policy> может использоваться пунктом PDP при принятии решения об авторизации.

В элемент <Policy> входит последовательность выборов между элементами <VariableDefinition> и <Rule>.

Правила, включенные в элемент <Policy>, должны объединяться с помощью алгоритма, установленного атрибутом RuleCombiningAlgId.

В элементе <Obligations> содержится набор обязательств, которые должны быть выполнены с помощью PEP вместе с решением об авторизации.

```
<xs:element name="Policy" type="xacml:PolicyType"/>
<xs:complexType name="PolicyType">
   <xs:sequence>
         <xs:element ref="xacml:Description" minOccurs="0"/>
         <xs:element ref="xacml:PolicyDefaults" minOccurs="0"/>
         <xs:element ref="xacml:CombinerParameters" minOccurs="0"/>
         <xs:element ref="xacml:Target"/>
         <xs:choice maxOccurs="unbounded">
               <xs:element ref="xacml:CombinerParameters" minOccurs="0"/>
               <xs:element ref="xacml:RuleCombinerParameters" minOccurs="0"/>
               <xs:element ref="xacml:VariableDefinition"/>
               <xs:element ref="xacml:Rule"/>
         </xs:choice>
         <xs:element ref="xacml:Obligations" minOccurs="0"/>
   </xs:sequence>
   <xs:attribute name="PolicyId" type="xs:anyURI" use="required"/>
   <xs:attribute name="Version" type="xacml:VersionType" default="1.0"/>
   <xs:attribute name="RuleCombiningAlgId" type="xs:anyURI" use="required"/>
</xs:complexType>
```

Элемент <Policy> является элементом составного типа **PolicyType**.

В элементе <Policy> содержатся следующие атрибуты и элементы:

– PolicyId [Обязательный]

  Идентификатор стратегического набора. В обязанности PAP входит гарантирование того, что никакие две стратегии, находящиеся в поле зрения PDP, не имеют один и тот же идентификатор. Этой цели можно добиться, следуя предписанной схеме URN или URI. Если идентификатор стратегического набора в виде URL, то это может быть отображено.

– Version [По умолчанию 1.0]

  Номер версии Policy.

– RuleCombiningAlgId [Обязательный]

  Идентификатор алгоритма объединения правил, с помощью которого должны объединяться компоненты <Policy>, <CombinerParameters>, <RuleCombinerParameters>.

– <Description> [Необязательный]

  Описание стратегии в произвольной форме.

– <PolicyDefaults> [Необязательный]

  Определяет набор значений по умолчанию, применимый к этой стратегии. Областью применения элемента <PolicyDefaults> должна быть включенная стратегия.

– <CombinerParameters> [Необязательный]

  Последовательность параметров, которая должна использоваться алгоритмом объединения правил.

– <RuleCombinerParameters> [Необязательный]

  Последовательность параметров, которая должна использоваться алгоритмом объединения правил.

– <Target> [Обязательный]

Элемент <Target> определяет применимость <Policy> к набору запросов о принятии решения.

Элемент <Target> может быть заявлен создателем <Policy> или он может быть вычислен из элементов <Target> упомянутых элементов <Rule>, либо методом дизъюнкции, либо методом объединения.

– <VariableDefinition> [Любое количество]

Распространенные определения функций, на которые могут ссылаться из любого места правила, где может быть найдено выражение.

– <Rule> [Любое количество]

Последовательность правил, которая должна объединяться в соответствии с атрибутом RuleCombiningAlgId. Должны учитываться те правила, элементы <Target> которых подходят для запроса о принятии решения. Те правила, элементы <Target> которых не подходят для запроса о принятии решения, не должны учитываться.

– <Obligations> [Необязательный]

Конъюнктивная последовательность обязательств, которые должны быть выполнены с помощью PEP вместе с решением об авторизации.

7.4.23 Элемент <PolicyDefaults>

Элемент <PolicyDefaults> должен устанавливать значения по умолчанию, которые применимы к элементу <Policy>.

```
<xs:element name="PolicyDefaults" type="xacml:DefaultsType"/>
<xs:complexType name="DefaultsType">
   <xs:sequence>
         <xs:choice>
               <xs:element ref="xacml:XPathVersion" minOccurs="0"/>
         </xs:choice>
   </xs:sequence>
</xs:complexType>
```

Элемент <PolicyDefaults> является элементом составного типа **DefaultsType**.

В элементе <PolicyDefaults> содержатся следующие элементы:

– <XPathVersion> [Необязательный]

Версия XPath по умолчанию.

7.4.24 Элемент <CombinerParameters>

Элемент <CombinerParameters> элемент переправляет параметры для алгоритма объединения правил или стратегий.

Если внутри одной и той же стратегии или стратегического набора появляется множество элементов <CombinerParameters>, то они должны быть приняты равными одному элементу <CombinerParameters>, содержащему сцепление всех последовательностей <CombinerParameters>, содержащихся во всех вышеупомянутых элементах <CombinerParameters> таким образом, чтобы порядок появления элементов <CominberParameters> сохранялся в сцеплении элементов <CombinerParameter>.

Заметим, что ни один из алгоритмов объединения, установленных в XACML 2.0, не записан в параметрической форме.

```
<xs:элемент name="CombinerParameters" type="xacml:CombinerParametersType"/>
<xs:complexType name="CombinerParametersType">
   <xs:sequence>
         <xs:element ref="xacml:CombinerParameter" minOccurs="0"
maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

Элемент <CombinerParameters> является элементом составного типа **CombinerParametersType**.

В элементе <CombinerParameters> содержатся следующие элементы:

– <CombinerParameter> [Любое количество]

Единый параметр.

Поддержка для элемента <CombinerParameters> является необязательной.

7.4.25 Элемент <CombinerParameter>

Элемент <CombinerParameter> передает единый параметр для алгоритма объединения правила или стратегии.

```
<xs:element name="CombinerParameter" type="xacml:CombinerParameterType"/>
<xs:complexType name="CombinerParameterType">
   <xs:sequence>
         <xs:element ref="xacml:AttributeValue"/>
   </xs:sequence>
   <xs:attribute name="ParameterName" type="xs:string" use="required"/>
</xs:complexType>
```

Элемент <CombinerParameter> является элементом составного типа **CombinerParameterType**.

В элементе <CombinerParameter> содержится следующий атрибут:

– ParameterName [Обязательный]

  Идентификатор параметра.

– AttributeValue [Обязательный]

  Значение параметра.

Поддержка для элемента <CombinerParameter> является необязательной.

7.4.26 Элемент <RuleCombinerParameters>

Элемент <RuleCombinerParameters> передает параметры, связанные с конкретным правилом внутри стратегии, для алгоритма объединения правил.

Каждый элемент <RuleCombinerParameters> должен быть связан с правилом, содержащимся внутри той же самой стратегии. Если множество элементов <RuleCombinerParameters> ссылаются на одно и то же правило, они должны быть приняты равными одному элементу <RuleCombinerParameters>, содержащему сцепление всех последовательностей <CombinerParameters>, содержащихся во всех вышеупомянутых элементах <RuleCombinerParameters>, таким образом, чтобы порядок появления элементов <RuleCominberParameters> сохранялся в сцеплении элементов <CombinerParameter>.

Заметим, что ни один из алгоритмов объединения правил, установленных в XACML 2.0, не записан в параметрической форме.

```
<xs:element name="RuleCombinerParameters"
type="xacml:RuleCombinerParametersType"/>
<xs:complexType name="RuleCombinerParametersType">
   <xs:complexContent>
         <xs:extension base="xacml:CombinerParametersType">
               <xs:attribute name="RuleIdRef" type="xs:string" use="required"/>
         </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

В элементе <RuleCombinerParameters> содержатся следующие элементы:

– RuleIdRef [Обязательный]

  Идентификатор <Rule>, содержащегося в данной стратегии.

Поддержка для элемента <RuleCombinerParameters> является необязательной, если только не реализуется поддержка для параметров объединителя.

7.4.27 Элемент <PolicyCombinerParameters>

Элемент <PolicyCombinerParameters> переносит параметры, связанные с конкретной стратегией внутри стратегического набора, для алгоритма объединения стратегий.

Каждый элемент <PolicyCombinerParameters> должен быть связан со стратегией, содержащейся внутри того же самого стратегического набора. Если множество элементов <PolicyCombinerParameters> ссылаются на одну и ту же стратегию, они должны быть приняты равными одному элементу <PolicyCombinerParameters>, содержащему сцепление всех последовательностей <CombinerParameters>, содержащихся во всех вышеупомянутых элементах <PolicyCombinerParameters>, таким образом, чтобы порядок появления элементов <PolicyCombinerParameters> сохранялся в сцеплении элементов <CombinerParameter>.

Заметим, что ни один из алгоритмов объединения стратегий, установленных в XACML 2.0, не записан в параметрической форме.

```
<xs:element name="PolicyCombinerParameters"
type="xacml:PolicyCombinerParametersType"/>
<xs:complexType name="PolicyCombinerParametersType">
   <xs:complexContent>
          <xs:extension base="xacml:CombinerParametersType">
                <xs:attribute name="PolicyIdRef" type="xs:anyURI"
use="required"/>
          </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

Элемент <PolicyCombinerParameters> является элементом составного типа **PolicyCombinerParametersType**.

В элементе <PolicyCombinerParameters> содержатся следующие элементы:

– PolicyIdRef [Обязательный]

  Идентификатор <Policy> или значение <PolicyIdReference>, содержащееся в стратегическом наборе.

Поддержка для элемента <PolicyCombinerParameters> является необязательной, если только не реализуется поддержка для параметров объединителя.

7.4.28 Элемент <PolicySetCombinerParameters>

Элемент <PolicySetCombinerParameters> передает параметры, связанные с конкретным стратегическим набором внутри стратегического набора для алгоритма объединения стратегий.

Каждый элемент <PolicySetCombinerParameters> должен быть связан со стратегическим набором, содержащимся внутри того же самого стратегического набора. Если множество элементов <PolicySetCombinerParameters> ссылаются на один и тот же стратегический набор, они должны быть приняты равными одному элементу <PolicySetCombinerParameters>, содержащему сцепление всех последовательностей <CombinerParameters>, содержащихся во всех вышеупомянутых элементах <PolicySetCombinerParameters>, таким образом, чтобы порядок появления элементов <PolicySetCombinerParameters> сохранялся в сцеплении элементов <CombinerParameter>.

Заметим, что ни один из алгоритмов объединения стратегий, установленных в XACML 2.0, не записан в параметрической форме.

```
<xs:element name="PolicySetCombinerParameters"
type="xacml:PolicySetCombinerParametersType"/>
<xs:complexType name="PolicySetCombinerParametersType">
   <xs:complexContent>
          <xs:extension base="xacml:CombinerParametersType">
                <xs:attribute name="PolicySetIdRef" type="xs:anyURI"
use="required"/>
          </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

Элемент <PolicySetCombinerParameters> является элементом составного типа **PolicySetCombinerParametersType**.

В элементе <PolicySetCombinerParameters> содержатся следующие элементы:

– PolicySetIdRef [Обязательный]

  Идентификатор <PolicySet> или значение <PolicySetIdReference>, содержащееся в стратегическом наборе.

Поддержка для элемента <PolicySetCombinerParameters> является необязательной, если только не реализуется поддержка для параметров объединителя.

7.4.29 Элемент <Rule>

Элемент <Rule> должен определять индивидуальные правила в стратегии. Основными компонентами этого элемента являются элементы <Target> и <Condition> и атрибут Effect.

Элемент <Rule> может оцениваться; в этом случае должна использоваться процедура оценки, определенная в пункте 7.6.9.

```
<xs:element name="Rule" type="xacml:RuleType"/>
<xs:complexType name="RuleType">
   <xs:sequence>
          <xs:element ref="xacml:Description" minOccurs="0"/>
```

```
            <xs:element ref="xacml:Target" minOccurs="0"/>
            <xs:element ref="xacml:Condition" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="RuleId" type="xs:string" use="required"/>
      <xs:attribute name="Effect" type="xacml:EffectType" use="required"/>
</xs:complexType>
```

Элемент <Rule> является элементом составного типа **RuleType**.

В элементе <Rule> содержатся следующие атрибуты и элементы:

– RuleId [Обязательный]

  Строка, идентифицирующая данное правило.

– Effect [Обязательный]

  Эффект правила. Значение этого атрибута либо "Permit", либо "Deny".

– <Description> [Необязательный]

  Описание правила в произвольной форме.

– <Target> [Необязательный]

  Определяет набор запросов о принятии решения, которые предназначены для оценки с помощью элемента <Rule>. Если этот элемент не включен, то цель для <Rule> должна определяться элементом <Target> включенного элемента <Policy>. Смотрите пункт 7.6.6 для более подробного описания.

– <Condition> [Необязательный]

  Предикат, который должен быть выполнен для этого правила, чтобы ему присвоить значение Effect.

7.4.30 EffectType простого типа

Простой тип **EffectType** определяет значения, разрешенные для атрибута Effect элемента <Rule> и для атрибута FulfillOn элемента <Obligation>.

```
<xs:simpleType name="EffectType">
      <xs:restriction base="xs:string">
            <xs:enumeration value="Permit"/>
            <xs:enumeration value="Deny"/>
      </xs:restriction>
</xs:simpleType>
```

7.4.31 Элемент <VariableDefinition>

Элемент <VariableDefinition> должен использоваться для определения значения, на которое может ссылаться элемент <VariableReference>. Имя, поддерживаемое для его атрибута VariableId не должно появляться в атрибуте VariableId любого другого элемента <VariableDefinition> внутри выполняемой стратегии. Элемент <VariableDefinition> может содержать неидентифицированный элемент <VariableReference>, но если это так, то соответствующий элемент <VariableDefinition> должен быть определен позже в выполняемой стратегии. Элементы <VariableDefinition> могут группироваться вместе или могут размещаться близко к ссылке в выполняемой стратегии. Может существовать ноль или более ссылок к каждому элементу <VariableDefinition>.

```
<xs:element name="VariableDefinition" type="xacml:VariableDefinitionType"/>
<xs:complexType name="VariableDefinitionType">
      <xs:sequence>
            <xs:element ref="xacml:Expression"/>
      </xs:sequence>
      <xs:attribute name="VariableId" type="xs:string" use="required"/>
</xs:complexType>
```

Элемент <VariableDefinition> является элементом составного типа **VariableDefinitionType**. У элемента <VariableDefinition> имеются следующие элементы и атрибуты:

– <Expression> [Обязательный]

  Любой элемент составного типа **ExpressionType**.

– `VariableId` [Обязательный]

Имя переменного определения.

7.4.32 Элемент <VariableReference>

Элемент `<VariableReference>` используется для ссылки на значение, определенное внутри того же самого выполняемого элемента `<Policy>`. Элемент `<VariableReference>` должен ссылаться на элемент `<VariableDefinition>` с помощью строкового уравнения со значениями их соответственных атрибутов `VariableId`. Должен существовать один и только один `<VariableDefinition>` внутри одного и того же выполняемого элемента `<Policy>`, на который ссылается `<VariableReference>`. Может существовать ноль или более элементов `<VariableReference>`, которые ссылаются на один и тот же элемент `<VariableDefinition>`.

```
<xs:element name="VariableReference" type="xacml:VariableReferenceType"
substitutionGroup="xacml:Expression"/>
<xs:complexType name="VariableReferenceType">
   <xs:complexContent>
          <xs:extension base="xacml:ExpressionType">
                 <xs:attribute name="VariableId" type="xs:string" use="required"/>
          </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

Элемент `<VariableReference>` является элементом составного типа **VariableReferenceType**, который является составным типом **ExpressionType** и членом группы подстановки элементов `<Expression>`. Элемент `<VariableReference>` может появляться в любом месте, где в схеме встречается элемент `<Expression>`.

В элементе `<VariableReference>` имеются следующие атрибуты:

– `VariableId` [Обязательный]

Имя, используемое для ссылки на значение, определенное в элементе `<VariableDefinition>`

элемент.

7.4.33 Элемент <Expression>

Элемент `<Expression>` не используется в стратегии напрямую. Элемент `<Expression>` означает, что элемент, который расширяет **ExpressionType** и является членом группы подстановки элементов `<Expression>`, должен появиться на его месте.

```
<xs:element name="Expression" type="xacml:ExpressionType" abstract="true"/>
<xs:complexType name="ExpressionType" abstract="true"/>
```

В группе подстановки элементов `<Expression>` имеются следующие элементы:

`<Apply>`, `<AttributeSelector>`, `<AttributeValue>`, `<Function>`, `<VariableReference>`, `<ActionAttributeDesignator>`, `<ResourceAttributeDesignator>`, `<SubjectAttributeDesignator>` и `<EnvironmentAttributeDesignator>`.

7.4.34 Элемент <Condition>

Элемент `<Condition>` является логической функцией по атрибутам субъекта, ресурса, действия и среды или по функциям атрибутов.

```
<xs:element name="Condition" type="xacml:ConditionType"/>
<xs:complexType name="ConditionType">
   <xs:sequence>
          <xs:element ref="xacml:Expression"/>
   </xs:sequence>
</xs:complexType>
```

В `<Condition>` содержится один элемент `<Expression>` с таким ограничением, что тип данных возвращения `<Expression>` должно быть "http://www.w3.org/2001/XMLSchema#boolean".

7.4.35 Элемент <Apply>

Элемент `<Apply>` обозначает приложение функции к ее аргументам, кодируя, таким образом, вызов функции. Элемент `<Apply>` может применяться к любой комбинации членов группы подстановки элементов `<Expression>`.

```
<xs:element name="Apply" type="xacml:ApplyType"
substitutionGroup="xacml:Expression"/>
<xs:complexType name="ApplyType">
```

```
   <xs:complexContent>
         <xs:extension base="xacml:ExpressionType">
               <xs:sequence>
                     <xs:element ref="xacml:Expression" minOccurs="0"
maxOccurs="unbounded"/>
               </xs:sequence>
               <xs:attribute name="FunctionId" type="xs:anyURI" use="required"/>
         </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

Элемент <Apply> является элементом составного типа **ApplyType**.

В элементе <Apply> содержатся следующие атрибуты и элементы:

– FunctionId [Обязательный]

  Идентификатор функции, которая должна применяться к аргументам. Функции, определенные в XACML, описаны в Приложении A.

– <Expression> [Необязательный]

  Аргументы функции, в которые могут входить другие функции.

7.4.36 Элемент <Function>

Элемент <Function> должен использоваться для именования функции, так как аргумент этой функции определен родительским элементом <Apply>. В том случае, если родительский элемент <Apply> является функцией "мешка" высшего порядка, именованная функция применяется к каждому элементу этого "мешка" или "мешков", идентифицированных в других аргументах этого родительского элемента.

```
<xs:element name="Function" type="xacml:FunctionType"
substitutionGroup="xacml:Expression"/>
<xs:complexType name="FunctionType">
   <xs:complexContent>
         <xs:extension base="xacml:ExpressionType">
               <xs:attribute name="FunctionId" type="xs:anyURI" use="required"/>
         </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

Элемент Function является элементом составного типа **FunctionType**.

В элементе Function содержатся следующие атрибуты:

– FunctionId [Обязательный]

  Идентификатор функции.

7.4.37 AttributeDesignatorType составного типа

Составной тип **AttributeDesignatorType** это тип для элементов, которые идентифицируют атрибуты по имени. В нем содержится информация, требуемая для сопоставления атрибутов в контексте запроса.

В нем также содержится информация для контроля поведения в случае, если в контексте не содержится ни одного подходящего атрибута.

Элементы этого типа не должны изменять семантику сопоставления именованных атрибутов, но могут сузить область поиска.

```
<xs:complexType name="AttributeDesignatorType">
   <xs:complexContent>
         <xs:extension base="xacml:ExpressionType">
               <xs:attribute name="AttributeId" type="xs:anyURI"
use="required"/>
               <xs:attribute name="DataType" type="xs:anyURI" use="required"/>
               <xs:attribute name="Issuer" type="xs:string" use="optional"/>
               <xs:attribute name="MustBePresent" type="xs:boolean"
use="optional" default="false"/>
         </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

Именованный атрибут должен быть сопоставим с атрибутом, если значения их соответственных атрибутов AttributeId, DataType и Issuer сопоставляются. AttributeId указателя атрибута должен сопоставляться, по равенству URI, с AttributeId этого атрибута. DataType указателя атрибута должен сопоставляться, по равенству URI, с DataType того же самого атрибута.

Если атрибут Issuer присутствует в указателе атрибута, то он должен сопоставляться, используя функцию "urn:oasis:names:tc:xacml:1.0:function:string-equal", с Issuer того же самого атрибута. Если Issuer не присутствует в указателе атрибута, то сопоставление этого атрибута с именованным атрибутом должно управляться только с помощью атрибутов AttributeId и DataType.

В <AttributeDesignatorType> содержатся следующие атрибуты:

– AttributeId [Обязательный]

  Этот атрибут должен устанавливать AttributeId, с которым должен сопоставляться атрибут.

– DataType [Обязательный]

  "Мешок", возвращаемый с помощью элемента <AttributeDesignator> должен содержать значения этого типа данных.

– Issuer [Необязательный]

  Этот атрибут, если он предоставляется, должен устанавливать Issuer, с которым сопоставляется атрибут.

– MustBePresent [Необязательный]

  Этот атрибут управляет событием возвращения "Indeterminate" или пустого "мешка" в таком событии, когда именованный атрибут отсутствует в контексте запроса.

7.4.38 Элемент <SubjectAttributeDesignator>

Этот элемент <SubjectAttributeDesignator> отыскивает "мешок" значений для именованного категоризированного атрибута субъекта из контекста запроса. Атрибут субъекта является атрибутом, который содержится внутри элемента <Subject> контекста запроса. Категоризированный субъект – это субъект, идентифицированный с помощью конкретного атрибута субъект-категория. Именованный категоризированный атрибут субъекта – это именованный атрибут субъекта для конкретного категоризированного субъекта.

Элемент <SubjectAttributeDesignator> должен возвращать "мешок", в котором содержатся все значения атрибута субъекта, которые сопоставляются с именованным категоризированным атрибутам субъекта. В случае, если в контексте не присутствует ни одного совпадающего атрибута, атрибут MustBePresent управляет возвращением пустого "мешка" или "Indeterminate".

SubjectAttributeDesignatorType расширяет семантику сопоставления **AttributeDesignatorType** таким образом, чтобы сузить область поиска атрибута до конкретного категоризированного субъекта, чтобы значение этого атрибута SubjectCategory элемента сопадало, по равенству URI, со значением атрибута SubjectCategory элемента <Subject> контекста запроса.

Если в контексте запроса содержится множество субъектов с тем же самым атрибутом SubjectCategory XML, то с ними нужно обращаться так, как если бы они были одним категоризированным субъектом.

<SubjectAttributeDesignator> может появиться в элементе <SubjectMatch> и может быть передан элементу <Apply> в качестве аргумента.

```
<xs:element name="SubjectAttributeDesignator"
type="xacml:SubjectAttributeDesignatorType"
substitutionGroup="xacml:Expression"/>
<xs:complexType name="SubjectAttributeDesignatorType">
   <xs:complexContent>
          <xs:extension base="xacml:AttributeDesignatorType">
                <xs:attribute name="SubjectCategory" type="xs:anyURI"
use="optional" default="urn:oasis:names:tc:xacml:1.0:subject-category:access-
subject"/>
          </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

Элемент <SubjectAttributeDesignator> является элементом типа **SubjectAttributeDesignatorType**. Составной тип **SubjectAttributeDesignatorType** расширяет составной тип **AttributeDesignatorType** с атрибутом SubjectCategory.

– SubjectCategory [Необязательный]

  Этот атрибут должен устанавливать категоризированный субъект, с которым должны сопоставляться именуемые атрибуты субъекта. Если SubjectCategory отсутствует, то должно использоваться его значение по умолчанию "urn:oasis:names:tc:xacml:1.0:subject-category:access-subject".

7.4.39 Элемент <ResourceAttributeDesignator>

Элемент <ResourceAttributeDesignator> отыскивает "мешок" значений для именованного атрибута ресурса из контекста запроса. Атрибут ресурса является атрибутом, содержащемся внутри элемента <Resource> контекста запроса. Именованный атрибут ресурса – это именованный атрибут, который сопоставляется с атрибутом ресурса. Именованный атрибут ресурса должен быть признан присутствующим, если существует хотя бы один атрибут ресурса, который сопоставляется с критерием, представленным ниже. Значение атрибута ресурса является значением атрибута, которое содержится внутри атрибута ресурса.

Элемент <ResourceAttributeDesignator> должен возвращать "мешок", в котором содержатся все значения атрибута ресурса, которые сопоставляются с именованным атрибутам ресурса. В случае, если в контексте не присутствует ни одного сопадающего атрибута, атрибут MustBePresent управляет возвращением пустого "мешка" или "Indeterminate".

SubjectAttributeDesignatorType расширяет семантику сопоставления **AttributeDesignatorType** таким образом, чтобы сузить область поиска атрибута до конкретного категоризированного субъекта, чтобы значение этого атрибута SubjectCategory элемента сопадало, по равенству URI, со значением атрибута SubjectCategory элемента <Subject> контекста запроса.

Именованный атрибут ресурса должен сопоставляться с атрибутом ресурса, как это установлено для каждого сопоставления по семантике в составном типе **AttributeDesignatorType**.

<ResourceAttributeDesignator> может появиться в элементе <ResourceMatch> и может быть передан элементу <Apply> в качестве аргумента.

```
<xs:element name="ResourceAttributeDesignator"
type="xacml:AttributeDesignatorType" substitutionGroup="xacml:Expression"/>
```

Элемент <ResourceAttributeDesignator> является элементом составного типа **AttributeDesignatorType**.

7.4.40 Элемент <ActionAttributeDesignator>

Элемент <ActionAttributeDesignator> отыскивает "мешок" значений для именованного атрибута действия из контекста запроса. Атрибут действия является атрибутом, содержащемся внутри элемента <Action> контекста запроса. У именованного атрибута действия имеются конкретные критерии (описанные ниже), по которым он сопоставляется с атрибутом действия. Именованный атрибут действия признается присутствующим, если существует хотя бы один атрибут действия, который сопоставляется с данными критериями. Значение атрибута действия является значением атрибута, которое содержится внутри атрибута действия.

Элемент <ActionAttributeDesignator> должен возвращать "мешок", в котором содержатся все значения атрибута действия, которые сопоставляются с именованным атрибутам действия. В случае, если в контексте не присутствует ни одного сопадающего атрибута, атрибут MustBePresent управляет возвращением пустого "мешка" или "Indeterminate".

SubjectAttributeDesignatorType расширяет семантику сопоставления **AttributeDesignatorType** таким образом, чтобы сузить область поиска атрибута до конкретного категоризированного субъекта, чтобы значение этого атрибута SubjectCategory элемента сопадало, по равенству URI, со значением атрибута SubjectCategory элемента <Subject> контекста запроса.

Именованный атрибут действия должен сопоставляться с атрибутом действия, как это установлено для каждого сопоставления по семантике в составном типе **AttributeDesignatorType**.

<ActionAttributeDesignator> может появиться в элементе <ActionMatch> и может быть передан элементу <Apply> в качестве аргумента.

```
<xs:element name="ActionAttributeDesignator" type="xacml:AttributeDesignatorType"
substitutionGroup="xacml:Expression"/>
```

Элемент <ActionAttributeDesignator> является элементом составного типа **AttributeDesignatorType**.

7.4.41 Элемент <EnvironmentAttributeDesignator>

Элемент <EnvironmentAttributeDesignator> отыскивает "мешок" значений для именованного атрибута среды из контекста запроса. Атрибут среды является атрибутом, содержащемся внутри элемента <Environment> контекста запроса. У именованного атрибута среды имеются конкретные критерии (описанные ниже), по которым он сопоставляется с атрибутом среды. Именованный атрибут среды признается присутствующим, если существует хотя бы один атрибут среды, который сопоставляется с данными критериями. Значение атрибута среды является значением атрибута, которое содержится внутри атрибута среды.

Элемент <EnvironmentAttributeDesignator> должен оценивать "мешок", в котором содержатся все значения атрибута среды, которые сопоставляются с именованным атрибутам среды. В случае, если в контексте не присутствует ни одного сопадающего атрибута, атрибут MustBePresent управляет возвращением пустого "мешка" или "Indeterminate".

Именованный атрибут среды должен сопоставляться с атрибутом среды, как это установлено для каждого сопоставления по семантике в составном типе **AttributeDesignatorType**.

<EnvironmentAttributeDesignator> может быть передан элементу <Apply> в качестве аргумента.

```
<xs:element name="EnvironmentAttributeDesignator"
type="xacml:AttributeDesignatorType" substitutionGroup="xacml:Expression"/>
```

Элемент <EnvironmentAttributeDesignator> является элементом составного типа **AttributeDesignatorType**.

7.4.42 Элемент <AttributeSelector>

Элемент <AttributeSelector> идентифицирует атрибуты по их местоположению в контексте запроса. Поддержка для элемента <AttributeSelector> является необязательной.

Атрибут RequestContextPath XML элемента <AttributeSelector> должен содержать легальное выражение XPath, узлом контекста которого является элемент <xacml-context:Request>. Элемент AttributeSelector должен оценивать "мешок" значений, чей тип данных устанавливается атрибутом DataType этого элемента. Если DataType установленный в AttributeSelector является простым типом данных (определенным в W3C Schema:2001, W3C Datatypes:2001, 3.2), то лексическое значение, возвращаемое выражением XPath, должно быть преобразовано в значение DataType, установленное в <AttributeSelector>. Если ошибка приводит к преобразованию значения, возвращаемого выражением Xpath, например, когда значение не является действительным экземпляром DataType, то значением элемента <AttributeSelector> должно быть "Indeterminate".

xs:string()
xs:boolean()
xs:integer()
xs:double()
xs:dateTime()
xs:date()
xs:time()
xs:hexBinary()
xs:base64Binary()
xs:anyURI()

Если DataType, установленный в AttributeSelector, не является одним из предшествующих простых DataTypes, то AttributeSelector должен возвратить "мешок" экземпляров установленного DataType. Если при преобразовании значений, возвращенных выражением XPath установленному DataType, происходит ошибка, то результатом AttributeSelector должно быть "Indeterminate".

Каждый узел, выбранный с помощью установленного выражения XPath, должен быть текстовым узлом, узлом атрибутов, узлом обработки команд или узлом комментариев. Строковое представление значения каждого узла должно быть преобразовано в значение атрибута, установленного типа данных, а результатом AttributeSelector является "мешок" значений атрибутов, созданных всеми выбранными узлами.

Если узел, выбранный с помощью установленного выражения XPath не является одним из тех, которые перечислены выше (то есть, текстовым узлом, узлом атрибутов, узлом обработки команд или узлом комментариев), то результатом прилагаемой стратегии должно быть "Indeterminate"со значением StatusCode - "urn:oasis:names:tc:xacml:1.0:status:syntax-error".

```
<xs:element name="AttributeSelector" type="xacml:AttributeSelectorType"
substitutionGroup="xacml:Expression"/>
<xs:complexType name="AttributeSelectorType">
   <xs:complexContent>
          <xs:extension base="xacml:ExpressionType">
                <xs:attribute name="RequestContextPath" type="xs:string"
use="required"/>
                <xs:attribute name="DataType" type="xs:anyURI" use="required"/>
                <xs:attribute name="MustBePresent" type="xs:boolean"
use="optional" default="false"/>
          </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

Элемент <AttributeSelector> является элементом составного типа **AttributeSelectorType**.

В элементе <AttributeSelector> имеются следующие атрибуты:

– RequestContextPath [Обязательный]

  Выражение XPath, чьим узлом контекста является элемент <xacml-context:Request>. Не должно быть ограничений на синтаксис XPath.

– `DataType` [Обязательный]

  В "мешке", возвращаемом элементом `<AttributeSelector>`, должны содержаться значения этого типа данных.

– `MustBePresent` [Необязательный]

  Этот атрибут управляет возвращением с помощью данного элемента "Indeterminate" или пустого "мешка" в случае, если ни один узел не будет выбран выражением XPath.

7.4.43 Элемент <AttributeValue>

В элементе `<xacml:AttributeValue>` должно содержаться буквенное значение атрибута.

```
<xs:element name="AttributeValue" type="xacml:AttributeValueType"
substitutionGroup="xacml:Expression"/>
<xs:complexType name="AttributeValueType" mixed="true">
   <xs:complexContent>
          <xs:extension base="xacml:ExpressionType">
                 <xs:sequence>
                        <xs:any namespace="##any" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
                 </xs:sequence>
                 <xs:attribute name="DataType" type="xs:anyURI" use="required"/>
                 <xs:anyAttribute namespace="##any" processContents="lax"/>
          </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

Элемент `<xacml:AttributeValue>` является элементом составного типа элемент **AttributeValueType**.

В элементе `<xacml:AttributeValue>` имеются следующие атрибуты:

– `DataType` [Обязательный]

  Тип данных этого значения атрибута.

7.4.44 Элемент <Obligations>

В элементе `<Obligations>` должен содержаться набор элементов `<Obligation>`.

Поддержка элемента `<Obligations>` является необязательной.

```
<xs:element name="Obligations" type="xacml:ObligationsType"/>
<xs:complexType name="ObligationsType">
   <xs:sequence>
          <xs:element ref="xacml:Obligation" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

Элемент `<Obligations>` является элементом составного типа **ObligationsType**.

В элементе `<Obligations>` содержится следующий элемент:

– `<Obligation>` [От одного до многих]

  Последовательность обязательств.

7.4.45 Элемент <Obligation>

В элементе `<Obligation>` должен содержаться идентификатор для обязательства и набор атрибутов, которые формируют аргументы действия, определенного этим обязательством. Атрибут `FulfillOn` должен показывать эффект, для получения которого должно выполняться это обязательство с помощью PEP.

```
<xs:element name="Obligation" type="xacml:ObligationType"/>
<xs:complexType name="ObligationType">
   <xs:sequence>
          <xs:element ref="xacml:AttributeAssignment" minOccurs="0"
maxOccurs="unbounded"/>
   </xs:sequence>
   <xs:attribute name="ObligationId" type="xs:anyURI" use="required"/>
   <xs:attribute name="FulfillOn" type="xacml:EffectType" use="required"/>
</xs:complexType>
```

Элемент <Obligation> является элементом составного типа **ObligationType**. Смотрите пункт 7.6.14 для ознакомления с описанием того, как определяется набор обязательств, который должен быть возвращен с помощью PDP.

В элементе <Obligation> содержатся следующие элементы и атрибуты:

– ObligationId [Обязательный]

  Идентификатор обязательства. Значение идентификатора обязательства должно быть интерпретировано с помощью PEP.

– FulfillOn [Обязательный]

  Эффект, для получения которого должно выполняться это обязательство с помощью PEP.

– <AttributeAssignment> [Необязательный]

  Назначение аргументов обязательства. Значения аргументов обязательства должны интерпретироваться с помощью PEP.

7.4.46 Элемент <AttributeAssignment>

Элемент <AttributeAssignment> используется для включения аргументов в обязательства. В нем должен содержаться AttributeId и соответствующее значение атрибута, с помощью расширения определения типа **AttributeValueType**. Элемент <AttributeAssignment> может использоваться любым способом, который согласуется с синтаксисом схемы, являющимся последовательностью элементов <xs:any>. Установленное значение должно восприниматься PEP, но оно не устанавливается более подробно языком XACML.

```
<xs:element name="AttributeAssignment" type="xacml:AttributeAssignmentType"/>
<xs:complexType name="AttributeAssignmentType" mixed="true">
   <xs:complexContent>
          <xs:extension base="xacml:AttributeValueType">
                 <xs:attribute name="AttributeId" type="xs:anyURI"
use="required"/>
          </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

Элемент <AttributeAssignment> является элементом составного типа **AttributeAssignmentType**.

В элементе <AttributeAssignment> содержатся следующие атрибуты:

– AttributeId [Обязательный]

  Идентификатор атрибута.

7.5 Синтаксис контекста

Фрагменты схемы в следующих пунктах являются ненормативными.

7.5.1 Элемент <Request>

Элемент <Request> является элементом высшего уровня в схеме контекста XACML. Элемент <Request> является уровнем абстракции, используемым языком стратегии. Для простоты выражений оценка стратегии в данной Рекомендации описывается в терминах операций над контекстом. Однако не требуется совместимый PDP для фактической иллюстрации этого контекста в форме документа XML. Но любая система, совместимая с XACML, должна вырабатывать точно такие же решения об авторизации, как если бы все входные значения были бы преобразованы в форму элемента <xacml-context:Request>.

В элементе <Request> содержатся элементы <Subject>, <Resource>, <Action> и <Environment>. В нем может быть множество элементов <Subject> и, при некоторых условиях, множество элементов <Resource>. В каждом элементе-потомке содержится последовательность элементов <xacml-context:Attribute>, связанных с этим субъектом, ресурсом, действием или средой, соответственно. Эти элементы <Attribute> могут формировать часть оценки стратегии.

```
<xs:element name="Request" type="xacml-context:RequestType"/>
<xs:complexType name="RequestType">
   <xs:sequence>
          <xs:element ref="xacml-context:Subject" maxOccurs="unbounded"/>
          <xs:element ref="xacml-context:Resource" maxOccurs="unbounded"/>
          <xs:element ref="xacml-context:Action"/>
          <xs:element ref="xacml-context:Environment"/>
   </xs:sequence>
</xs:complexType>
```

Элемент <Request> является элементом составного типа **RequestType**.

В элементе <Request> содержатся следующие элементы:

– <Subject> [От одного до многих]

  Устанавливает информацию о субъекте контекста запроса, перечисляя последовательность элементов <Attribute>, связанных с этим субъектом. Разрешен один или более элементов <Subject>. Субъект является объектом, связанным с запросом о доступе. Например, один субъект может представлять человека-пользователя, который инициировал приложение, из которого поступил запрос; другой субъект может представлять рабочую программу приложения, ответственную за создание запроса; еще один субъект может представлять устройство, на котором выполнялось это приложение; и еще один субъект может представлять объект, который должен быть получателем ресурса. Атрибуты каждого из этих объектов должны включаться в отдельные элементы <Subject>.

– <Resource> [От одного до многих]

  Устанавливает информацию о ресурсе или ресурсах, для которых запрашивается доступ, перечисляя последовательности элементов <Attribute>, связанных с этим ресурсом. В него может входить элемент <ResourceContent>.

– <Action> [Обязательный]

  Устанавливает требуемое действие, которое должно выполняться над ресурсом, перечисляя набор элементов <Attribute>, связанных с этим действием.

– <Environment> [Обязательный]

  Содержит набор элементов <Attribute> для этой среды.

7.5.2 Элемент <Subject>

Элемент <Subject> устанавливает субъект, перечисляя последовательность элементов <Attribute>, связанных с этим субъектом.

```
<xs:element name="Subject" type="xacml-context:SubjectType"/>
<xs:complexType name="SubjectType">
   <xs:sequence>
         <xs:element ref="xacml-context:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
   </xs:sequence>
   <xs:attribute name="SubjectCategory" type="xs:anyURI"
default="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"/>
</xs:complexType>
```

Элемент <Subject> является элементом составного типа **SubjectType**.

В элементе <Subject> содержатся следующие элементы и атрибуты:

– SubjectCategory [Необязательный]

  Этот атрибут показывает роль, которую родительский элемент <Subject> играет в формировании запроса о доступе. Если этот атрибут не представлен в заданном элементе <Subject>, то должно использоваться значение по умолчанию "urn:oasis:names:tc:xacml:1.0:subject-category:access-subject", показывающее, что родительский элемент <Subject> представляет объект, ответственный, в конечном счете, за инициацию запроса о доступе.

  Если атрибут "urn:oasis:names:tc:xacml:2.0:subject-category" с одним и тем же значением содержится в более чем одном элементе <Subject>, то PDP должен обращаться с содержанием этих элементов, как если бы они содержались в одном и том же элементе <Subject>.

– <Attribute> [Любое количество]

  Последовательность атрибутов, которые применяются к этому субъекту.

В типичном случае элемент <Subject> будет содержать <Attribute> с AttributeId атрибута "urn:oasis:names:tc:xacml:1.0:subject:subject-id", содержащим идентичность этого субъекта.

В элементе <Subject> могут содержаться дополнительные элементы <Attribute>.

7.5.3 Элемент <Resource>

Элемент <Resource> устанавливает информацию о ресурсе, к которому запрашивается доступ, перечисляя последовательность элементов <Attribute>, связанных с этим ресурсом. В него может входить содержание ресурса.

```
<xs:element name="Resource" type="xacml-context:ResourceType"/>
<xs:complexType name="ResourceType">
   <xs:sequence>
          <xs:element ref="xacml-context:ResourceContent" minOccurs="0"/>
          <xs:element ref="xacml-context:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

Элемент <Resource> является элементом составного типа **ResourceType**.

В элементе <Resource> содержатся следующие элементы:

– <ResourceContent> [Необязательный]

  Содержание ресурса.

– <Attribute> [Любое количество]

  Последовательность атрибутов ресурса.

В элементе <Resource> может содержаться один или более элементов <Attribute> с AttributeId атрибута "urn:oasis:names:tc:xacml:2.0:resource:resource-id". Каждый такой <Attribute> должен быть абсолютным и полностью разрешенным представлением идентичности отдельного ресурса, к которому запрашивается доступ, и если установлен любой <Attribute> с этим AttributeId, то должен быть установлен <Attribute> для каждого такого индивидуального представления идентичности ресурса. Все такие элементы <Attribute> должны относиться к одному и тому же отдельному экземпляру ресурса. Профиль для конкретного ресурса может устанавливать отдельное нормативное представление для экземпляров этого ресурса; в этом случае любой <Attribute> с таким AttributeId должен использовать только одно это представление.

В элементе <Resource> могут содержаться дополнительные элементы <Attribute>.

7.5.4 Элемент <ResourceContent>

Элемент <ResourceContent> является воображаемым символом-заполнителем для содержания данного ресурса. Если стратегия XACML ссылается на содержание этого ресурса с помощью элемента <AttributeSelector> элемент, то элемент <ResourceContent> должен быть включен в строку RequestContextPath.

```
<xs:complexType name="ResourceContentType" mixed="true">
   <xs:sequence>
          <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
   </xs:sequence>
   <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:complexType>
```

Элемент <ResourceContent> является элементом составного типа **ResourceContentType**.

Элемент <ResourceContent> позволяет наличие случайных элементов и атрибутов.

7.5.5 Элемент <Action>

Элемент <Action> устанавливает требуемое действие над ресурсом, перечисляя набор элементов <Attribute>, связанных с этим действием.

```
<xs:element name="Action" type="xacml-context:ActionType"/>
<xs:complexType name="ActionType">
   <xs:sequence>
          <xs:element ref="xacml-context:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

Элемент <Action> является элементом составного типа **ActionType**.

В элементе <Action> содержатся следующие элементы:

– <Attribute> [Любое количество]

  Список атрибутов действия, которые должны быть выполнены над этим ресурсом.

7.5.6 Элемент <Environment>

В элементе <Environment> содержится набор атрибутов этой среды.

```
<xs:element name="Environment" type="xacml-context:EnvironmentType"/>
<xs:complexType name="EnvironmentType">
   <xs:sequence>
         <xs:element ref="xacml-context:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

Элемент <Environment> является элементом составного типа **EnvironmentType**.

В элементе <Environment> содержатся следующие элементы:

– <Attribute> [Любое количество]

  Список атрибутов среды. Атрибутами среды являются атрибуты, которые не связаны ни с ресурсом, ни с действием, ни с любым из субъектов запроса о доступе.

7.5.7 Элемент <Attribute>

Элемент <Attribute> является центральной абстракцией контекста запроса. В нем содержится атрибут метаданных и один или более значений атрибутов. В метаданных атрибута содержится идентификатор атрибута и запрашивающая сторона атрибута. Элементы <AttributeDesignator> и <AttributeSelector> в стратегии могут ссылаться на атрибуты посредством этих метаданных.

```
<xs:element name="Attribute" type="xacml-context:AttributeType"/>
<xs:complexType name="AttributeType">
   <xs:sequence>
         <xs:element ref="xacml-context:AttributeValue" maxOccurs="unbounded"/>
   </xs:sequence>
   <xs:attribute name="AttributeId" type="xs:anyURI" use="required"/>
   <xs:attribute name="DataType" type="xs:anyURI" use="required"/>
   <xs:attribute name="Issuer" type="xs:string" use="optional"/>
</xs:complexType>
```

Элемент <Attribute> является элементом составного типа **AttributeType**.

В элементе <Attribute> содержатся следующие атрибуты и элементы:

– AttributeId [Обязательный]

  Идентификатор атрибута. Языком XACML зарезервировано несколько идентификаторов для обозначения наиболее часто используемых атрибутов.

– DataType [Обязательный]

  Тип данных содержания элемента <xacml-context:AttributeValue>. Это должен быть или простой тип, определенный этой Рекомендацией, или типом (простым или структурированным), определенным в пространстве имен, заявленном в элементе <xacml-context>.

– Issuer [Необязательный]

  Сторона, запрашивающая атрибут. Например, значение этого атрибута может быть x500Name, которое связано с открытым ключом, или может быть каким-то другим идентификатором, которым обмениваются вне полосы запрашивающая и отвечающая стороны.

– <xacml-context:AttributeValue> [От одного до многих]

  Одно или более значений атрибута. В каждом значении атрибута может быть пустое содержание, попадающийся один раз или много раз.

7.5.8 Элемент <AttributeValue>

В элементе <xacml-context:AttributeValue> содержится значение атрибута.

```
<xs:element name="AttributeValue" type="xacml-context:AttributeValueType"/>
<xs:complexType name="AttributeValueType" mixed="true">
   <xs:sequence>
         <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
   </xs:sequence>
   <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:complexType>
```

Элемент <xacml-context:AttributeValue> является элементом составного типа **AttributeValueType**.

Тип данных <xacml-context:AttributeValue> должен устанавливаться с помощью использования атрибута DataType родительского элемента <Attribute>.

7.5.9 Элемент <Response>

Элемент <Response> является элементом высшего уровня в схеме контекста XACML. Элемент <Response> является уровнем абстракции, используемым языком стратегии. Любая частная система, использующая XACML, должна преобразовывать элемент <Response> контекста XACML в форму своего решения об авторизации.

Элемент <Response> инкапсулирует решение об авторизации, выпущенное пунктом PDP. В него включена последовательность одного или более результатов, с одним элементом <Result> на каждый запрашиваемый ресурс. Множество результатов могут возвращаться некоторыми реализациями, в особенности теми, которые поддерживают профиль XACML для запросов для множества ресурсов. Поддержка для множества результатов является необязательной.

```
<xs:element name="Response" type="xacml-context:ResponseType"/>
<xs:complexType name="ResponseType">
   <xs:sequence>
         <xs:element ref="xacml-context:Result" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

Элемент <Response> является элементом составного типа **ResponseType**.

В элементах <Response> содержатся следующие элементы:

– <Result> [От одного до многих]

  Результат решения об авторизации.

7.5.10 Элемент <Result>

Элемент <Result> представляет результат решения об авторизации, установленный атрибутом ResourceId. В него может входить набор обязательств, которые должны быть выполнены с помощью PEP. Если PEP не воспринимает или не может выполнить обязательство, то он должен действовать так, как если бы PDP отказал в доступе к запрашиваемому ресурсу.

```
<xs:complexType name="ResultType">
   <xs:sequence>
         <xs:element ref="xacml-context:Decision"/>
         <xs:element ref="xacml-context:Status" minOccurs="0"/>
         <xs:element ref="xacml:Obligations" minOccurs="0"/>
   </xs:sequence>
   <xs:attribute name="ResourceId" type="xs:string" use="optional"/>
</xs:complexType>
```

Элемент <Result> является элементом составного типа **ResultType**.

В элементе <Result> содержатся следующие атрибуты и элементы:

– ResourceId [Необязательный]

  Идентификатор запрашиваемого ресурса. Если этот атрибут пропущен, то идентичность ресурса такая, как установлено атрибутом ресурса "urn:oasis:names:tc:xacml:1.0:resource:resource-id" в соответствующем элементе <Request>.

– <Decision> [Обязательный]

  Решение об авторизации: "Permit", "Deny", "Indeterminate" или "NotApplicable".

– <Status> [Необязательный]

Показывает, была ли допущена ошибка во время оценки запроса о принятии решения и, необязательно, информацию об этих ошибках. Если в элементе <Response> содержатся элементы <Result>, в которых все элементы <Status> идентичны и элемент <Response> содержится в упаковщике протоколов, который может переправлять информацию о состоянии, то информация об общем состоянии может быть помещена в этот упаковщик протоколов и данный элемент <Status> может быть пропущен во всех элементах <Result>.

– <Obligations> [Необязательный]

Список обязательств, которые должны быть выполнены с помощью PEP. Если PEP не воспринимает или не может выполнить обязательство, то он должен действовать так, как если бы PDP отказал в доступе к запрашиваемому ресурсу.

7.5.11 Элемент <Decision>

В элементе <Decision> содержится результат оценки стратегии.

```
<xs:element name="Decision" type="xacml-context:DecisionType"/>
<xs:simpleType name="DecisionType">
   <xs:restriction base="xs:string">
          <xs:enumeration value="Permit"/>
          <xs:enumeration value="Deny"/>
          <xs:enumeration value="Indeterminate"/>
          <xs:enumeration value="NotApplicable"/>
   </xs:restriction>
</xs:simpleType>
```

Элемент <Decision> является элементом простого типа **DecisionType**.

Значения элемента <Decision> могут быть следующими:

– Permit: Запрашиваемый доступ разрешен.

– Deny: В запрашиваемом доступе отказано.

– Indeterminate: PDP не в состоянии оценить запрашиваемый доступ. Причинами такой несостоятельности могут быть: пропущенные атрибуты, сетевые ошибки во время поиска стратегии, деление на ноль во время оценки стратегии, ошибки синтаксиса в запросе о принятии решения или в стратегии и т. д.

– NotApplicable: В PDP отсутствует стратегия, применимая к данному запросу о принятии решения.

7.5.12 Элемент <Status>

Элемент <Status> представляет состояние результата решения об авторизации.

```
<xs:element name="Status" type="xacml-context:StatusType"/>
<xs:complexType name="StatusType">
   <xs:sequence>
          <xs:element ref="xacml-context:StatusCode"/>
          <xs:element ref="xacml-context:StatusMessage" minOccurs="0"/>
          <xs:element ref="xacml-context:StatusDetail" minOccurs="0"/>
   </xs:sequence>
</xs:complexType>
```

Элемент <Status> является элементом составного типа элемент **StatusType**.

В элементе <Status> содержатся следующие элементы:

– <StatusCode> [Обязательный]

Код состояния.

– <StatusMessage> [Необязательный]

Сообщение о состоянии, описывающее код состояния.

– <StatusDetail> [Необязательный]

Дополнительная информация о состоянии.

7.5.13 Элемент <StatusCode>

В элементе <StatusCode> содержится значение основного кода состояния и необязательная последовательность второстепенных кодов состояния.

```
<xs:element name="StatusCode" type="xacml-context:StatusCodeType"/>
<xs:complexType name="StatusCodeType">
   <xs:sequence>
          <xs:element ref="xacml-context:StatusCode" minOccurs="0"/>
   </xs:sequence>
   <xs:attribute name="Value" type="xs:anyURI" use="required"/>
</xs:complexType>
```

Элемент <StatusCode> является элементом составного типа **StatusCodeType**.

В элементе <StatusCode> содержатся следующие атрибуты и элементы:

- Value [Обязательный]

  Смотрите B.8 для ознакомления со списком значений.

- <StatusCode> [Любое количество]

  Второстепенный код состояния. Этот код состояния квалифицирует код состояния родителя.

7.5.14 Элемент <StatusMessage>

Элемент <StatusMessage> является описанием кода состояния в произвольной форме.

```
<xs:element name="StatusMessage" type="xs:string"/>
```

Элемент <StatusMessage> является элементом типа **xs:string**.

7.5.15 Элемент <StatusDetail>

Элемент <StatusDetail> квалифицирует элемент <Status> с помощью дополнительной информации.

```
<xs:element name="StatusDetail" type="xacml-context:StatusDetailType"/>
<xs:complexType name="StatusDetailType">
   <xs:sequence>
          <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

Элемент <StatusDetail> является элементом составного типа элемент **StatusDetailType**.

В элементе <StatusDetail> разрешается наличие случайного содержания XML.

Включение элемента <StatusDetail> является необязательным. Однако, если PDP возвращает одно из следующих, определенных XACML, значений <StatusCode> и включает элемент <StatusDetail>, то применяются следующие правила.

```
urn:oasis:names:tc:xacml:1.0:status:ok
```

Пункту PDP запрещено возвращать элемент <StatusDetail> вместе со значением состояния "ok".

```
urn:oasis:names:tc:xacml:1.0:status:missing-attribute
```

В PDP может быть выбор: не возвращать никакую информацию <StatusDetail> или возвращать элемент <StatusDetail>, содержащий один или более элементов <xacml-context: MissingAttributeDetail>.

```
urn:oasis:names:tc:xacml:1.0:status:syntax-error
```

Пункту PDP запрещено возвращать элемент <StatusDetail> вместе со значением состояния "syntax-error". Ошибка синтаксиса может представлять либо проблему из-за используемой стратегии, либо из-за контекста запроса. Пункт PDP может возвращать <StatusMessage>, описывающий эту проблему.

```
urn:oasis:names:tc:xacml:1.0:status:processing-error
```

PDP запрещено возвращать элемент <StatusDetail> вместе со значением состояния "processing-error". Этот код состояния показывает на внутреннюю проблему в PDP. По соображениям безопасности пункт PDP может не возвращать никакую дополнительную информацию в пункт PEP. В случае появления ошибки, связанной с делением на ноль или другой вычислительной ошибки, PDP может возвратить <StatusMessage>, описывающий природу этой ошибки.

7.5.16 Элемент <MissingAttributeDetail>

Элемент `<MissingAttributeDetail>` переправляет информацию об атрибутах, требующихся для оценки этой стратегии, которые были пропущены в контексте запроса.

```
<xs:element name="MissingAttributeDetail" type="xacml-
context:MissingAttributeDetailType"/>
<xs:complexType name="MissingAttributeDetailType">
<xs:sequence>
<xs:element ref="xacml-context:AttributeValue" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute name="AttributeId" type="xs:anyURI" use="required"/>
<xs:attribute name="DataType" type="xs:anyURI" use="required"/>
   <xs:attribute name="Issuer" type="xs:string" use="optional"/>
</xs:complexType>
```

Элемент `<MissingAttributeDetail>` является элементом составного типа **MissingAttributeDetailType**.

В элементе `<MissingAttributeDetal>` содержатся следующие атрибуты и элементы:

- `AttributeValue` [Необязательный]

  Требуемое значение пропущенного атрибута.

- `<AttributeId>` [Обязательный]

  Идентификатор пропущенного атрибута.

- `<DataType>` [Обязательный]

  Тип данных пропущенного атрибута.

- `Issuer` [Необязательный]

  Этот атрибут, если он предоставляется, должен устанавливать требуемый `Issuer` пропущенного атрибута.

Если пунктом PDP включаются элементы `<xacml-context:AttributeValue>` в элемент `<MissingAttributeDetail>`, то это показывает на допустимые значения для данного атрибута. Если не включено ни одного элемента `<xacml-context:AttributeValue>`, то это показывает имена атрибутов, которые PDP не удалось разрешить во время его оценки. Список атрибутов может быть частичным или полным. Отсутствуют гарантии того, что предоставление с помощью PDP отсутствующих значений или атрибутов будет достаточно для выполнения стратегии.

7.6 Функциональные требования XACML

В этом пункте устанавливаются некоторые функциональные требования, которые не связаны напрямую с созданием или потреблением конкретного элемента XACML.

7.6.1 Пункт осуществления стратегии

В этом пункте описываются требования к PEP.

Прикладные функции в той роли PEP, когда он защищает набор ресурсов и запрашивает решение об авторизации у PDP. Пункт PEP должен выполнять решения об авторизации, созданные PDP.

7.6.1.1 База PEP

Если решением является "Permit", то PEP должен разрешить доступ. Если решение сопровождается обязательствами, то PEP должен разрешить доступ только если он воспринимает, в состоянии выполнить и будет выполнять эти обязательства.

Если решением является "Deny", то PEP должен отказать в доступе. Если решение сопровождается обязательствами, то PEP должен отказать в доступе только если он воспринимает, в состоянии выполнить и будет выполнять эти обязательства.

Если решением является "Not Applicable", то поведение PEP не определено.

Если решением является "Indeterminate", то поведение PEP не определено.

7.6.1.2 PEP со смещением в сторону отказа

Если решением является "Permit", то PEP должен разрешить доступ. Если решение сопровождается обязательствами, то PEP должен разрешить доступ только если он воспринимает, в состоянии выполнить и будет выполнять эти обязательства.

Все остальные решения приведут к отказу в доступе.

ПРИМЕЧАНИЕ. – Другие действия, например, консультация дополнительных PDP, переформулировка/повторное представление запроса о принятии решения и т. д., не запрещены.

7.6.1.3 PEP со смещением в сторону разрешения доступа

Если решением является "Deny", то PEP должен отказать в доступе. Если решение сопровождается обязательствами, то PEP должен отказать в доступе только если он воспринимает, в состоянии выполнить и будет выполнять эти обязательства.

Все остальные решения приведут к разрешению доступа.

ПРИМЕЧАНИЕ. – Другие действия, например, консультация дополнительных PDP, переформулировка/повторное представление запроса о принятии решения и т. д., не запрещены.

7.6.2 Оценка атрибутов

Атрибуты представляются в контексте запроса обработчиком контекста, независимо от того, появляются они или нет в первоначальном запросе о принятии решения, и на них ссылаются в стратегии с помощью указателей атрибутов и селекторов атрибутов субъекта, ресурса, действия или среды. Именуемый атрибут – это термин, использующийся для таких критериев, которые конкретные указатели атрибутов и селекторы атрибутов субъекта, ресурса, действия и среды используют для ссылки на конкретные атрибуты в элементах контекста запроса субъекта, ресурса, действия и среды, соответственно.

7.6.2.1 Структурированные атрибуты

Элементы `<xacml:AttributeValue>` и `<xacml-context:AttributeValue>` могут содержать экземпляр структурированного типа данных XML, например `<ds:KeyInfo>`. В данной Рекомендации поддерживаются несколько способов для сравнения содержания таких элементов.

1) В некоторых случаях такие элементы можно сравнивать, используя одну из строковых функций XACML, такую как "string-regexp-match", описанную ниже. При этом требуется, чтобы элементу был задан тип данных "http://www.w3.org/2001/XMLSchema#string". Например, структурированный тип данных, который фактически имеет вид **ds:KeyInfo/KeyName**, появился бы в контексте, как:

```
<AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
   &lt;ds:KeyName&gt;jhibbert-key&lt;/ds:KeyName&gt;
</AttributeValue>
```

   В общем случае, этот метод пригоден, если только структурированный тип данных довольно простой.

2) Элемент `<AttributeSelector>` может использоваться для выбора содержания подэлемента листа структурированного типа данных посредством выражения XPath. Затем это значение можно сравнивать, используя одну из поддерживаемых функций XACML, подходящую для его простого типа данных. Этому методу требуется поддержка со стороны PDP для необязательного свойства выражений XPath.

3) Элемент `<AttributeSelector>` может использоваться для выбора любого узла в структурированном типе данных посредством выражения XPath. Затем этот узел можно сравнивать, используя одну из функций на основе XPath, описанную в пункте A.3. Для этого метода требуется поддержка со стороны PDP для необязательных выражений XPath и свойств функций XPath.

7.6.2.2 "Мешки" атрибутов

В языке XACML определяется неявная совокупность его типов данных. XACML ссылается к совокупности значений, которые являются отдельным типом данных, как к "мешку". "Мешки" типов данных нужны потому, что совокупности узлов от ресурса XML или контекста запроса XACML могут возвращать более одного значения.

Элемент `<AttributeSelector>` использует выражение XPath для установления выбора данных от ресурса XML. Результат выражения XPath обозначает набор узлов, в котором содержатся все узлы листов из ресурса XML, которые сопоставимы с предикатом в выражении XPath. Основываясь на различных функциях индексации, предоставляемых в W3C XPath:1999, предполагается, что результирующим набором узлов является совокупность сопоставимых узлов. В данной Рекомендации также описывается элемент `<AttributeDesignator>`, для того чтобы иметь ту же самую методологию сопоставления для атрибутов в контексте запроса XACML.

Значения в "мешке" не упорядочены и некоторые значения могут повторяться. Не должно быть понятия о "мешке", содержащем "мешки", или о "мешке", содержащем значения разных типов (т. е. "мешок" в XACML должен содержать только значения одного и того же типа данных).

7.6.2.3 Многозначные атрибуты

Если отдельный элемент `<Attribute>` в контексте запроса содержит множество дочерних элементов `<xacml-context:AttributeValue>`, то "мешок" значений, образующийся в результате оценки элемента `<Attribute>`, должен быть идентичен "мешку" значений, образовавшемся в результате оценки контекста, в котором каждый

элемент <xacml-context:AttributeValue> появляется в отдельном элементе <Attribute>, и каждый несет идентичные метаданные.

7.6.2.4 Сопоставление атрибутов

В именованные атрибуты включены особые критерии, с помощью которых происходит сопоставление атрибутов в контексте. Атрибут устанавливает AttributeId и DataType, а именованный атрибут также устанавливает Issuer. Именованный атрибут должен быть сопоставим с атрибутом, если значения их соответственных атрибутов AttributeId, DataType и необязательного Issuer сопоставимы внутри их конкретных элементов – субъекта, ресурса, действия или среды – данного контекста. Атрибут AttributeId именованного атрибута должен быть сопоставим, по равенству URI, с AttributeId соответствующего атрибута контекста. Атрибут DataType именованного атрибута должен быть сопоставим, по равенству URI, с DataType соответствующего атрибута контекста. Если Issuer предоставляется в именованном атрибуте, то он должен быть сопоставим, используя функцию urn:oasis:names:tc:xacml:1.0:function:string-equal, с Issuer соответствующего атрибута контекста. Если Issuer не предоставляется в именованном атрибуте, то сопоставление атрибута контекста с именованным атрибутом должно управляться только атрибутами AttributeId и DataType, независимо от присутствия, отсутствия или фактического значения Issuer в соответствующем атрибуте контекста. В случае селектора атрибутов сопоставление атрибута с именованным атрибутом должно управляться выражением XPath и DataType.

7.6.2.5 Поиск атрибутов

PDP должен запрашивать значения атрибутов в контексте запроса у обработчика контекста. PDP должен ссылаться на атрибуты, как если бы они были физическим документом контекста запроса, но обработчик контекста несет ответственность за получение и поставку запрошенных значений любыми подходящими способами. Обработчик контекста должен возвращать значения атрибутов, сопоставимые с указателем атрибутов или селектором атрибутов, и формировать "мешки" значений с установленным типом данных.Если ни один атрибут из контекста запроса не сопоставим, то такой атрибут считается пропущенным. Если атрибут пропущен, то MustBePresent управляет возвращением результата пустого "мешка" или"Indeterminate", либо с помощью указателя атрибутов, либо с помощью селектора атрибутов либо. Если MustBePresent имеет значение "False" (значение по умолчанию), то результатом пропущенного атрибута будет пустой "мешок". Если MustBePresent имеет значение "True", то результатом пропущенного атрибута будет "Indeterminate". Этот результат "Indeterminate" должен быть обработан в соответствии со спецификацией выполняющихся выражений, правил, стратегий и стратегических наборов. Если результатом является "Indeterminate", то AttributeId, DataType и Issuer этого атрибута могут быть перечислены в решении об авторизации. Однако у PDP есть выбор не возвращать такую информацию по соображениям безопасности.

7.6.2.6 Атрибуты среды

Если значение для одного из этих атрибутов доставляется в запросе о принятии решения, то обработчик контекста должен использовать это значение. Иначе, обработчик контекста должен доставить значение. В случае атрибутов даты и времени, предоставляемое значение должно иметь такую семантику "дата и время, которые применяются к запросу о принятии решения".

7.6.3 Оценка выражений

В языке XACML устанавливаются выражения в терминах тех элементов, которые перечислены ниже, из которых элементы <Apply> и <Condition> рекурсивно составляют выражения большего размера. Действительные значения должны быть правильного типа; это означает, что типы каждого элемента, содержащиеся внутри элементов <Apply> и <Condition> должны согласовываться с соответственными типами аргументов той функции, которая именована с помощью атрибута FunctionId. Результирующий тип элемента <Apply> или <Condition> должен быть результирующим типом функции, который может быть сужен до простого типа данных, или "мешка" простого типа данных, с помощью унификации типа. В языке XACML определяется результат оценки "Indeterminate", о котором сказано, что он является результатом недействительного выражения или операционной ошибки, произошедшей во время оценки выражения.

В языке XACML определяются эти элементы, как элементы, которые должны быть в группе подстановки элемента <Expression>:

- <xacml:AttributeValue>
- <xacml:SubjectAttributeDesignator>
- <xacml:ResourceAttributeDesignator>
- <xacml:ActionAttributeDesignator>
- <xacml:EnvironmentAttributeDesignator>
- <xacml:AttributeSelector>
- <xacml:Apply>
- <xacml:Condition>
- <xacml:Function>
- <xacml:VariableReference>

7.6.4 Арифметическая оценка

В документе IEEE 754 установлено, как оценивать арифметические функции в контексте; в нем установлены значения по умолчанию для точности, округления и т. п. В языке XACML должна использоваться эта спецификация для проведения оценки всех целых функций и функций с двойной точностью, полагаясь на Расширенный контекст значений по умолчанию (Extended Default Context), с улучшенной двойной точностью:

- флаги: все установить в 0;
- активаторы ловушек все установить в 0, за исключением активатора ловушки "деление на ноль", который должен быть установлен в 1;
- точность: устанавливается в назначенную двойную точность;
- округление: устанавливается в round-half-even (округление к ближайшему целому, среднее значение округляется к четному числу).

7.6.5 Оценка сопоставлений

Элементы сопоставления атрибутов появляются в элементе `<Target>` правил, стратегий и стратегического набора. Такими элементами являются следующие:

- `<SubjectMatch>`
- `<ResourceMatch>`
- `<ActionMatch>`
- `<EnvironmentMatch>`

Эти элементы представляют логические выражения по атрибутам субъекта, ресурса, действия и среды, соответственно. В элементе сопоставления содержится атрибут `MatchId`, устанавливающий функцию, которую нужно использовать при выполнении оценки сопоставлений, элемент `<xacml:AttributeValue>` и `<AttributeDesignator>` или `<AttributeSelector>`, устанавливающий атрибут в контексте, который должен быть сопоставлен с установленным значением.

Атрибут `MatchId` должен устанавливать функцию, которая сравнивает два аргумента, возвращая тип результата "http://www.w3.org/2001/XMLSchema#boolean". Значение атрибута, установленное в элементе сопоставления, должно быть предоставлено функции `MatchId` в качестве ее первого аргумента. Элемент "мешка", возвращенный элементом `<AttributeDesignator>` или `<AttributeSelector>`, должен быть предоставлен функции `MatchId` в качестве ее второго аргумента, как поясняется ниже. `DataType` атрибута `<xacml:AttributeValue>` должен быть сопоставим с типом данных первого аргумента, который ожидается функцией `MatchId`. `DataType` элемента `<AttributeDesignator>` или `<AttributeSelector>` должен быть сопоставим с типом данных второго аргумента, ожидаемого функцией `MatchId`.

Стандартными функциями XACML, которые отвечают требованиям для использования в качестве значения атрибута `MatchId`, являются:

```
urn:oasis:names:tc:xacml:2.0:function:-type-equal
urn:oasis:names:tc:xacml:2.0:function:-type-greater-than
urn:oasis:names:tc:xacml:2.0:function:-type-greater-than-or-equal
urn:oasis:names:tc:xacml:2.0:function:-type-less-than
urn:oasis:names:tc:xacml:2.0:function:-type-less-than-or-equal
urn:oasis:names:tc:xacml:2.0:function:-type-match
```

Дополнительно, функции, которые находятся строго внутри расширения к XACML, могут появиться, как значение для атрибута `MatchId`, и эти функции могут использовать типы данных, которые тоже являются расширениями, поскольку функция расширения возвращает логический результат и принимает два отдельных базовых типа в качестве своих входных данных. Функция, используемая в качестве значения для атрибута `MatchId`, должна быть легко индексируемой. Использование неиндексируемых или сложных функций может помешать действенной оценке запросов о принятии решения.

Семантика оценки для элемента сопоставления должна быть следующей. Если во время оценки элемента `<AttributeDesignator>` или `<AttributeSelector>` появляется операционная ошибка, то результатом всего выражения должно быть "Indeterminate". Если элемент `<AttributeDesignator>` или `<AttributeSelector>` был оценен, как пустой "мешок", то результатом этого выражения должно быть "False". Иначе, функция `MatchId` должна применяться между `<xacml:AttributeValue>` и каждым элементом "мешка", возвращенного от элемента `<AttributeDesignator>` или `<AttributeSelector>`. Если по крайней мере одно из этих применений функции будет оценено, как "True", то результатом всего выражения должно быть "True". Иначе, если по крайней мере одно из этих приложений функции дает в результате "Indeterminate", то результатом должно быть "Indeterminate". Наконец, если все приложения функции оцениваются как "False", то результатом всего выражения должно быть "False".

Также возможно выразить семантику элемента сопоставления цели при каком-нибудь условии. Например, выражение сопоставления цели, которое сравнивает "имя субъекта", начинающееся с имени "John", может быть выражено следующим образом:

```
<SubjectMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-regexp-match">
  <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
    John.*
  </AttributeValue>
```

```
  <SubjectAttributeDesignator
     AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
     DataType="http://www.w3.org/2001/XMLSchema#string"/>
</SubjectMatch>
```

С другой стороны, та же самая семантика сопоставления может быть выражена, как элемент <Apply> при каком-нибудь условии, с помощью использования функции "urn:oasis:names:tc:xacml:1.0:function:any-of", следующим образом:

```
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
  <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-regexp-match"/>
  <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
    John.*
  </AttributeValue>
  <SubjectAttributeDesignator
     AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
     DataType="http://www.w3.org/2001/XMLSchema#string"/>
</Apply>
```

7.6.6 Оценка цели

Значением цели должно быть "Match", если субъекты, ресурсы, действия и среды, установленные в цели, все сопоставимы со значениями в контексте запроса. Если у любого из субъектов, ресурсов, действий и сред, установленных в цели, значение "Indeterminate", то эта цель должна быть "Indeterminate". Иначе, цель должна быть "No match". Таблица сопоставлений для цели показана в таблице 7-1.

Таблица 7-1/X.1142 – Таблица сопоставлений для цели

Значение субъектов	Значение ресурсов	Значение действий	Значение сред	Значение цели
"Match"	"Match"	"Match"	"Match"	"Match"
"No match"	"Match" or "No match"	"Match" or "No match"	"Match" or "No match"	"No match"
"Match" or "No match"	"No match"	"Match" or "No match"	"Match" or "No match"	"No match"
"Match" or "No match"	"Match" or "No match"	"No match"	"Match" or "No match"	"No match"
"Match" or "No match"	"Match" or "No match"	"Match" or "No match"	"No match"	"No match"
"Indeterminate"	Don't care (Не важно)	Don't care (Не важно)	Don't care (Не важно)	"Indeterminate"
Don't care (Не важно)	"Indeterminate"	Don't care (Не важно)	Don't care (Не важно)	"Indeterminate"
Don't care (Не важно)	Don't care (Не важно)	"Indeterminate"	Don't care (Не важно)	"Indeterminate"
Don't care (Не важно)	Don't care (Не важно)	Don't care (Не важно)	"Indeterminate"	"Indeterminate"

Субъекты, ресурсы, действия и среды должны быть сопоставимы со значениями в контексте запроса, если по крайней мере один из их элементов <Subject>, <Resource>, <Action> или <Environment>, соответственно, сопоставим со значением в контексте запроса. Таблица сопоставления для субъектов показана в таблице 7-2. Таблицы сопоставления для ресурсов, действий и сред являются аналогичными.

Таблица 7-2/X.1142 – Таблица сопоставления для субъектов

<Subject> **значения**	<Subjects> **значение**
По крайней мере одно "Match"	"Match"
Нет совпадений и, по крайней мере, одно "Indeterminate"	"Indeterminate"
Все "No match"	"No match"

Субъект, ресурс, действие или среда должна быть сопоставима со значением в контексте запроса, если значением всех его элементов <SubjectMatch>, <ResourceMatch>, <ActionMatch> или <EnvironmentMatch>, соответственно, является "True".

Таблица сопоставления для субъекта показана в таблице 7-3. Таблицы сопоставления для ресурса, действия и среды являются аналогичными.

Таблица 7-3/X.1142 – Таблица сопоставления для субъекта

<SubjectMatch> **значения**	<Subject> **значение**
Все "True"	"Match"
Нет "False" и, по крайней мере, один "Indeterminate"	"Indeterminate"
По крайней мере один "False"	"No match"

7.6.7 Оценка VariableReference

Элемент <VariableReference> дает ссылку на отдельный элемент<VariableDefinition>, содержащийся внутри того же самого элемента <Policy>. Элемент <VariableReference>, который не дает ссылку на конкретный элемент <VariableDefinition> внутри выполняемого элемента <Policy>, называется неопределенной ссылкой. Стратегии с неопределенной ссылкой являются недействительными.

В любом месте, где появляется <VariableReference>, возникает такой эффект, как если бы текст элемента <Expression>, определенный в элементе <VariableDefinition>, заменил элемент <VariableReference>. Допустима любая схема оценки, которая сохраняет эту семантику. Например, выражение в элементе <VariableDefinition> может быть оценено определенным значением и помещено в кэш-память для многократных ссылок без последствий (т. е. значение элемента <Expression> остается одним и тем же для всей оценки стратегии). Эта характеристика является одним из преимуществ языка XACML, как декларативного языка.

7.6.8 Оценка условия

Значение условия должно быть "True", если отсутствует элемент <Condition>, или если он оценивается как "True". Ее значение должно быть "False", если элемент <Condition> оценивается как "False". Значение условия должно быть "Indeterminate", если выражение, заключенное в элементе <Condition>, оценивается как "Indeterminate."

7.6.9 Оценка правила

У правила есть значение, которое может быть вычислено с помощью оценки содержания. В оценку правила входит отдельная оценка цели правила и условия. Таблица истинности правила показана в таблице 7-4.

Таблица 7-4/X.1142 – Таблица истинности правила

Цель	Условие	Значение правила
"Match"	"True"	Effect
"Match"	"False"	"NotApplicable"
"Match"	"Indeterminate"	"Indeterminate"
"No-match"	Don't care (Не важно)	"NotApplicable"
"Indeterminate"	Don't care (Не важно)	"Indeterminate"

Если значением цели является "No-match" или "Indeterminate", то значением правила должно быть "NotApplicable" или "Indeterminate", соответственно, безотносительно к значению условия. Таким образом, для этих случаев нет необходимости в оценке условия.

Если значением цели является "Match" и значением условия является "True", то эффект, определенный в прилагающемся элементе <Rule> должен определить значение правила.

7.6.10 Оценка стратегии

Значение стратегии должно определяться только ее содержанием, рассматриваемым в связи с содержанием контекста запроса. Значение стратегии должно определяться с помощью оценки цели стратегии и правил.

Цель стратегии должна оцениваться для определения применимости данной стратегии. Если цель оценивается как "Match", то значение стратегии должно определяться с помощью оценки правил стратегии, в соответствии с установленным алгоритмом объединения правил. Если цель оценивается как "No-match", то значением стратегии должно быть "NotApplicable". Если цель оценивается как "Indeterminate", то значением стратегии должно быть "Indeterminate".

Таблица истинности стратегии показана в таблице 7-5.

Таблица 7-5/X.1142 – Таблица истинности стратегии

Цель	Значения правила	Значение стратегии
"Match"	По крайней мере одним значением правила является его Effect	Устанавливается алгоритмом объединения правил
"Match"	Все значения правила являются "NotApplicable"	"NotApplicable"
"Match"	По крайней мере одним значением правила является "Indeterminate"	Устанавливается алгоритмом объединения правил
"No-match"	Don't care (Не важно)	"NotApplicable"
"Indeterminate"	Don't care (Не важно)	"Indeterminate"

Значение правила "По крайней мере одним значением правила является его "Effect" означает, что элемент `<Rule>` отсутствует или одно или более правил, содержащихся в этой стратегии, применимы к запросу о принятии решения (т. е. он возвращает значение его "Effect"). Значение правил "Все значения правила являются 'NotApplicable'" должно использоваться, если ни одно из правил, содержащихся в стратегии, не применимо к запросу и, если ни одно из правил, содержащихся в стратегии, не возвращает значение "Indeterminate". Если ни одно правило, содержащееся в стратегии, не применимо к запросу, но одно или более правил возвращает значение "Indeterminate", то правила должны оцениваться значением " По крайней мере одним значением правила является "Indeterminate".

Если значением цели является "No-match" или "Indeterminate", то значением стратегии должно быть "NotApplicable" или "Indeterminate", соответственно, безотносительно к значению правил. Таким образом, для этих случаев нет необходимости в оценке правил.

Если значением цели является "Match", а значением правила является " По крайней мере одним значением правила является его "Effect" или "По крайней мере одним значением правила является "Indeterminate", то алгоритм объединения правил, установленный в этой стратегии, должен определять значение стратегии.

Заметим, что ни один из алгоритмов объединения правил, определенных в данной Рекомендации, не принимает параметров. Однако нестандартные алгоритмы объединения могут принимать параметры. В таком случае, значения этих параметров, связанные с правилами, должны приниматься во внимание при оценке стратегии. Параметры и их типы должны быть определены в спецификации алгоритмов объединения. Если определенная реализация поддерживает параметры объединителя и, если параметры объединителя присутствуют в стратегии, то значения параметров должны быть предоставлены для реализации алгоритма объединения.

7.6.11 Оценка стратегического набора

Значение стратегического набора должно определяться его содержанием, рассматриваемом в связи с содержанием контекста запроса. Значение стратегического набора должно определяться с помощью оценки цели стратегического набора, стратегий и стратегического набора в соответствии с установленным алгоритмом объединения стратегий.

Цель стратегического набора должна оцениваться для определения применимости данного стратегического набора. Если цель оценивается как "Match", то значение стратегического набора должно определяться с помощью оценки стратегий стратегического набора и стратегических наборов, в соответствии с установленным алгоритмом объединения стратегий. Если цель оценивается как "No-match", то значением стратегического набора должно быть "NotApplicable". Если цель оценивается как "Indeterminate", то значением стратегического набора должно быть "Indeterminate".

Таблица истинности стратегического набора показана в таблице 7-6.

Таблица 7-6/X.1142 – Таблица истинности стратегического набора

Цель	Значения стратегии	Значение стратегического набора
"Match"	По крайней мере одно значение стратегии является его Decision	Устанавливается алгоритмом объединения стратегий
"Match"	Все значения стратегии являются "NotApplicable"	"NotApplicable"
"Match"	По крайней мере одним значением стратегии является "Indeterminate"	Устанавливается алгоритмом объединения стратегий
"No-match"	Don't care (Не важно)	"NotApplicable"
"Indeterminate"	Don't care (Не важно)	"Indeterminate"

Значение стратегий "По крайней мере одним значением стратегии является его "Decision"" должно использоваться, если отсутствуют стратегии или стратегические наборы, которые содержатся в данном стратегическом наборе или на которые дана ссылка в данном стратегическом наборе, а также если одна или более стратегий или стратегических наборов содержится в данном стратегическом наборе либо на нее дана ссылка в данном стратегическом наборе, является применимой к запросу о принятии решения (т. е. возвращает значение, определенное с помощью ее алгоритма объединения). Значение стратегий "Все значения стратегии являются "NotApplicable" должно использоваться, если ни одно из правил, содержащихся в стратегии, не применимо к запросу и если ни одна стратегия или стратегический набор, содержащаяся в данном стратегическом наборе или на которую дана ссылка в данном стратегическом наборе, не применима к этому запросу, и если ни одна стратегия или стратегический набор, содержащаяся в данном стратегическом наборе или на которую дана ссылка в данном стратегическом наборе, не возвращает значение "Indeterminate". Если ни одна стратегия или стратегический набор, содержащаяся в данном стратегическом наборе или на которую дана

ссылка в данном стратегическом наборе, не применима к запросу, но одна или более стратегий или стратегических наборов возвращает значение "Indeterminate", то эти стратегии должны оцениваться значением "По крайней мере одним значением стратегии является "Indeterminate".

Если значением цели является "No-match" или "Indeterminate", то значением стратегического набора должно быть "NotApplicable" или "Indeterminate", соответственно, безотносительно к значению стратегий. Таким образом, для этих случаев нет необходимости в оценке стратегий.

Если значением цели является "Match", а значением стратегий является "По крайней мере одним значением стратегии является его "Decision"" или "По крайней мере одним значением стратегии является "Indeterminate", то алгоритм объединения стратегий, установленный в этом стратегическом наборе, должен определять значение стратегического набора.

Заметим, что ни один из алгоритмов объединения стратегий, определенных в XACML 2.0, не принимает параметров. Однако нестандартные алгоритмы объединения могут принимать параметры. В таком случае, значения этих параметров, связанные со стратегиями, должны приниматься во внимание при оценке стратегических наборов. Параметры и их типы должны быть определены в спецификации алгоритмов объединения. Если определенная реализация поддерживает параметры объединителя и, если параметры объединителя присутствуют в стратегии, то значения параметров должны быть предоставлены для реализации алгоритма объединения.

7.6.12 Иерархические ресурсы

Часто встречаются случаи иерархической организации ресурса (например, система файлов, документ XML). В языке XACML предоставлено несколько необязательных механизмов для поддержания иерархических ресурсов, как это рассматривается в данной Рекомендации.

7.6.13 Решение об авторизации

Относительно конкретного запроса о принятии решения PDP определен с помощью алгоритма объединения стратегий и стратегического набора и/или наборов стратегий. Пункт PDP должен возвратить контекст ответа, как если бы он оценил отдельный стратегический набор, состоящий из этого алгоритма объединения стратегий и стратегического набора и/или наборов стратегий.

Пункт PDP должен оценить стратегический набор, как это установлено в пунктах 7.4 и 7.6. Пункт PDP должен возвратить контекст ответа с одним элементом `<Decision>`, имеющим значение "Permit", "Deny", "Indeterminate" или "NotApplicable".

Если PDP не может принять решение, то должен быть возвращен элемент `<Decision>` со значением "Indeterminate" `<Decision>`.

7.6.14 Обязательства

В стратегии или стратегическом наборе может содержаться одно или более обязательств. При оценке такой стратегии или стратегического набора обязательство должно передаваться следующему уровню оценки (прилагающейся или упоминающейся стратегии, стратегическому набору или решению об авторизации), только если эффект оцениваемой стратегии или стратегического набора сопоставим со значением атрибута `FulfillOn` данного обязательства.

Как следствие этой процедуры, ни одно обязательство не должно быть возвращено в PEP, если стратегии или стратегические наборы, из которых они извлечены, не оцениваются или если результатом их оценки является "Indeterminate" или "NotApplicable", или если решение, являющееся итогом оценки стратегии или стратегического набора, не сопоставимо с решением, являющимся итогом оценки введенного стратегического набора.

Если рассматривать оценку PDP как дерево стратегических наборов или стратегий, каждая из которых возвращает "Permit" или "Deny", то в набор обязательств, возвращаемых пунктом PDP в пункт PEP, будут включены только обязательства, связанные с такими путями, в которых эффект на каждом уровне оценки тот же, что и эффект, возвращаемый пунктом PDP. В тех ситуациях, когда недопустима любая неопределенность, должен использоваться детерминированный алгоритм объединения, такой как ordered-deny-overrides (упорядоченный запрет замещения).

7.6.15 Обработка исключений

В языке XACML установлено поведение PDP в следующих ситуациях.

7.6.15.1 Неподдерживаемая функциональность

Если PDP пытается оценить стратегический набор или стратегию, содержащую дополнительный тип элемента или функции, который не поддерживается пунктом PDP, то PDP должен возвращать значение "Indeterminate" элемента `<Decision>`. Если также возвращается элемент `<StatusCode>`, то его значением должно быть "`urn:oasis:names:tc:xacml:1.0:status:syntax-error`" в случае неподдерживаемого типа элемента, и "`urn:oasis:names:tc:xacml:1.0:status:processing-error`" в случае неподдерживаемой функции.

7.6.15.2 Ошибки типа и синтаксиса

Если стратегия, содержащая недействительный синтаксис, оценивается пунктом PDP XACML в тот момент, когда получен запрос о принятии решения, то результатом этой стратегии должно быть "Indeterminate" со значением `StatusCode`:

```
"urn:oasis:names:tc:xacml:1.0:status:syntax-error"
```

Если стратегия, содержащая недействительные статические типы данных, оценивается пунктом PDP XACML в тот момент, когда получен запрос о принятии решения, то результатом этой стратегии должно быть "Indeterminate" со значением StatusCode:

```
"urn:oasis:names:tc:xacml:1.0:status:processing-error"
```

7.6.15.3 Пропущенные атрибуты

Отсутствие сопоставимых атрибутов в контексте запроса для любого из указателей или селекторов атрибута, которые обнаружены в данной стратегии, должны привести к значению "Indeterminate" элемента <Decision>. Если в этом случае предоставлен и код состояния, то должно использоваться значение:

```
"urn:oasis:names:tc:xacml:1.0:status:missing-attribute"
```

чтобы показать, что для принятия окончательного решения требуется дополнительная информация. В таком случае элемент <Status> может перечислять имена и типы данных любых атрибутов субъектов, ресурсов, действий и сред, которые требуются для PDP, чтобы уточнить его решение. Пункт PEP может переутверждать уточненный контекст запроса в ответ на содержание "Indeterminate" элемента <Decision> с кодом статуса:

```
"urn:oasis:names:tc:xacml:1.0:missing-attribute"
```

добавляя значения атрибутов для имен атрибутов, которые были перечислены в предыдущем ответе. Если PDP возвращает содержание "Indeterminate" элемента <Decision> с кодом статуса:

```
"urn:oasis:names:tc:xacml:1.0:missing-attribute"
```

то ему запрещено перечислять имена и типы данных любого атрибута субъекта, ресурса, действия или среды, для которых значения были предоставлены в первоначальном запросе. Заметим, что это требование вынуждает PDP, в конечном счете, возвращать решение об авторизации "Permit", "Deny" или "Indeterminate" с каким-либо другим кодом статуса в ответ на успешно уточненные запросы.

7.7 Пункты расширяемости XACML

В этом пункте описаны пункты внутри модели XACML и схема, в которую добавлены расширения. Данный пункт является информативным.

7.7.1 Расширяемые типы атрибутов XML

У следующих атрибутов XML имеются значения, которые являются идентификаторами URI. Они могут быть расширены созданием новых идентификаторов URI, связанных с новой семантикой для этих атрибутов.

- AttributeId;
- DataType;
- FunctionId;
- MatchId;
- ObligationId;
- PolicyCombiningAlgId;
- RuleCombiningAlgId;
- StatusCode;
- SubjectCategory.

7.7.2 Структурированные атрибуты

В элементах <xacml:AttributeValue> и <xacml-context:AttributeValue> может содержаться экземпляр структурированного типа данных XML. Здесь перечислено несколько методов, которым требуются расширения XACML.

1) Для заданного структурированного типа данных сообщество пользователей XACML может определить новые идентификаторы атрибутов для каждого подэлемента листа структурированного типа данных, тип которого согласуется с одним из, определенных языком XACML, простых типов данных. Используя эти новые идентификаторы атрибутов, пункты PEP или обработчики контекста, используемые этим сообществом пользователей, могут выровнять экземпляры этого структурированного типа данных в последовательность отдельных элементов <Attribute. Каждый такой элемент <Attribute> можно сравнивать, используя функции, определенные XACML. Используя этот метод, сам структурированный тип данных никогда не появляется в элементе <xacml-context:AttributeValue>.

2) Сообщество пользователей XACML может определить новую функцию, которая может использоваться для сравнения значения этого структурированного типа данных с каким-то другим значением. Этот метод может быть использован только теми PDP, которые поддерживают новую функцию.

7.8 Совместимость

Языком XACML определяется несколько функций, у которых есть какое-то особое применение, вот почему они не заявлены, как поддерживающиеся функции в реализации, которая заявлена как совместимая с данной Рекомендацией.

В этом пункте перечисляются те части данной Рекомендации, которые должны быть включены в реализацию пункта PDP, который заявлен, как совместимый с XACML v2.0.

ПРИМЕЧАНИЕ. – "M" означает обязательный к реализации. "O" означает необязательный.

7.8.1 Элементы схемы

Данная реализация должна поддерживать те элементы схемы, которые помечены "M".

Наименование элемента	M/O
xacml-context:Action	M
xacml-context:Attribute	M
xacml-context:AttributeValue	M
xacml-context:Decision	M
xacml-context:Environment	M
xacml-context:MissingAttributeDetail	M
xacml-context:Obligations	O
xacml-context:Request	M
xacml-context:Resource	M
xacml-context:ResourceContent	O
xacml-context:Response	M
xacml-context:Result	M
xacml-context:Status	M
xacml-context:StatusCode	M
xacml-context:StatusDetail	O
xacml-context:StatusMessage	O
xacml-context:Subject	M
xacml:Action	M
xacml:ActionAttributeDesignator	M
xacml:ActionMatch	M
xacml:Actions	M
xacml:Apply	M
xacml:AttributeAssignment	O
xacml:AttributeSelector	O
xacml:AttributeValue	M
xacml:CombinerParameters	O
xacml:CombinerParameter	O
xacml:Condition	M
xacml:Description	M
xacml:Environment	M
xacml:EnvironmentMatch	M
xacml:EnvironmentAttributeDesignator	M
xacml:Environments	M
xacml:Expression	M
xacml:Function	M
xacml:Obligation	O
xacml:Obligations	O
xacml:Policy	M
xacml:PolicyCombinerParameters	O

Наименование элемента	M/O
xacml:PolicyDefaults	O
xacml:PolicyIdReference	M
xacml:PolicySet	M
xacml:PolicySetDefaults	O
xacml:PolicySetIdReference	M
xacml:Resource	M
xacml:ResourceAttributeDesignator	M
xacml:ResourceMatch	M
xacml:Resources	M
xacml:Rule	M
xacml:RuleCombinerParameters	O
xacml:Subject	M
xacml:SubjectMatch	M
xacml:Subjects	M
xacml:Target	M
xacml:VariableDefinition	M
xacml:VariableReference	M
xacml:XPathVersion	O

7.8.2 Префиксы идентификатора

Следующие префиксы идентификатора зарезервированы языком XACML.

Идентификатор
urn:oasis:names:tc:xacml:2.0
urn:oasis:names:tc:xacml:2.0:conformance-test
urn:oasis:names:tc:xacml:2.0:context
urn:oasis:names:tc:xacml:2.0:example
urn:oasis:names:tc:xacml:1.0:function
urn:oasis:names:tc:xacml:2.0:function
urn:oasis:names:tc:xacml:2.0:policy
urn:oasis:names:tc:xacml:1.0:subject
urn:oasis:names:tc:xacml:1.0:resource
urn:oasis:names:tc:xacml:1.0:action
urn:oasis:names:tc:xacml:1.0:environment
urn:oasis:names:tc:xacml:1.0:status

7.8.3 Алгоритмы

В данную реализацию должны быть включены алгоритмы объединения стратегий и правил, связанные со следующими идентификаторами, помеченными "M".

Алгоритм	M/O
urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides	M
urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides	M
urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides	M
urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:permit-overrides	M
urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-applicable	M
urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:first-applicable	M
urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:only-one-applicable	M
urn:oasis:names:tc:xacml:1.1:rule-combining-algorithm:ordered-deny-overrides	M
urn:oasis:names:tc:xacml:1.1:policy-combining-algorithm:ordered-deny-overrides	M
urn:oasis:names:tc:xacml:1.1:rule-combining-algorithm:ordered-permit-overrides	M
urn:oasis:names:tc:xacml:1.1:policy-combining-algorithm:ordered-permit-overrides	M

7.8.4 Коды состояний

Поддержка реализации для элемента <StatusCode> является необязательной, но если этот элемент поддерживается, то следующие коды состояний должны поддерживаться и должны использоваться так, как это установлено языком XACML.

Идентификатор	M/O
urn:oasis:names:tc:xacml:1.0:status:missing-attribute	M
urn:oasis:names:tc:xacml:1.0:status:ok	M
urn:oasis:names:tc:xacml:1.0:status:processing-error	M
urn:oasis:names:tc:xacml:1.0:status:syntax-error	M

7.8.5 Атрибуты

Данная реализация должна поддерживать атрибуты, связанные со следующими идентификаторами, как установлено языком XACML. Если значения этих атрибутов не присутствуют в запросе о принятии решения, то их значения должны быть предоставлены обработчиком контекста. Поэтому, в отличие от большинства других атрибутов, их семантика не прозрачна для PDP.

Идентификатор	M/O
urn:oasis:names:tc:xacml:1.0:environment:current-time	M
urn:oasis:names:tc:xacml:1.0:environment:current-date	M
urn:oasis:names:tc:xacml:1.0:environment:current-dateTime	M

7.8.6 Идентификаторы

Данная реализация должна использовать атрибуты, связанные со следующими идентификаторами так, как это установлено языком XACML. Это требование относится в первую очередь к реализациям PAP или PEP, которые используют XACML, так как семантика этих атрибутов прозрачна для PDP.

Идентификатор	M/O
urn:oasis:names:tc:xacml:1.0:subject:authn-locality:dns-name	O
urn:oasis:names:tc:xacml:1.0:subject:authn-locality:ip-address	O
urn:oasis:names:tc:xacml:1.0:subject:authentication-method	O
urn:oasis:names:tc:xacml:1.0:subject:authentication-time	O
urn:oasis:names:tc:xacml:1.0:subject:key-info	O
urn:oasis:names:tc:xacml:1.0:subject:request-time	O
urn:oasis:names:tc:xacml:1.0:subject:session-start-time	O
urn:oasis:names:tc:xacml:1.0:subject:subject-id	O
urn:oasis:names:tc:xacml:1.0:subject:subject-id-qualifier	O
urn:oasis:names:tc:xacml:1.0:subject-category:access-subject	M
urn:oasis:names:tc:xacml:1.0:subject-category:codebase	O
urn:oasis:names:tc:xacml:1.0:subject-category:intermediary-subject	O
urn:oasis:names:tc:xacml:1.0:subject-category:recipient-subject	O
urn:oasis:names:tc:xacml:1.0:subject-category:requesting-machine	O
urn:oasis:names:tc:xacml:1.0:resource:resource-location	O
urn:oasis:names:tc:xacml:1.0:resource:resource-id	M
urn:oasis:names:tc:xacml:1.0:resource:simple-file-name	O
urn:oasis:names:tc:xacml:1.0:action:action-id	O
urn:oasis:names:tc:xacml:1.0:action:implied-action	O

7.8.7 Типы данных

Данная реализация должна поддерживать типы данных, связанные со следующими идентификаторами, помеченными "M".

Тип данных	M/O
http://www.w3.org/2001/XMLSchema#string	M
http://www.w3.org/2001/XMLSchema#boolean	M
http://www.w3.org/2001/XMLSchema#integer	M
http://www.w3.org/2001/XMLSchema#double	M
http://www.w3.org/2001/XMLSchema#time	M

Тип данных	M/O
http://www.w3.org/2001/XMLSchema#date	M
http://www.w3.org/2001/XMLSchema#dateTime	M
urn:oasis:names:tc:xacml:2.0:data-types:dayTimeDuration	M
urn:oasis:names:tc:xacml:2.0:data-types:yearMonthDuration	M
http://www.w3.org/2001/XMLSchema#anyURI	M
http://www.w3.org/2001/XMLSchema#hexBinary	M
http://www.w3.org/2001/XMLSchema#base64Binary	M
urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name	M
urn:oasis:names:tc:xacml:1.0:data-type:x500Name	M

7.8.8 Функции

Данная реализация должна правильно обрабатывать те функции, которые связаны с идентификаторами, помеченными "M".

Функция	M/O
urn:oasis:names:tc:xacml:1.0:function:string-equal	M
urn:oasis:names:tc:xacml:1.0:function:boolean-equal	M
urn:oasis:names:tc:xacml:1.0:function:integer-equal	M
urn:oasis:names:tc:xacml:1.0:function:double-equal	M
urn:oasis:names:tc:xacml:1.0:function:date-equal	M
urn:oasis:names:tc:xacml:1.0:function:time-equal	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-equal	M
urn:oasis:names:tc:xacml:1.0:function:dayTimeDuration-equal	M
urn:oasis:names:tc:xacml:1.0:function:yearMonthDuration-equal	M
urn:oasis:names:tc:xacml:1.0:function:anyURI-equal	M
urn:oasis:names:tc:xacml:1.0:function:x500Name-equal	M
urn:oasis:names:tc:xacml:1.0:function:rfc822Name-equal	M
urn:oasis:names:tc:xacml:1.0:function:hexBinary-equal	M
urn:oasis:names:tc:xacml:1.0:function:base64Binary-equal	M
urn:oasis:names:tc:xacml:1.0:function:integer-add	M
urn:oasis:names:tc:xacml:1.0:function:double-add	M
urn:oasis:names:tc:xacml:1.0:function:integer-subtract	M
urn:oasis:names:tc:xacml:1.0:function:double-subtract	M
urn:oasis:names:tc:xacml:1.0:function:integer-multiply	M
urn:oasis:names:tc:xacml:1.0:function:double-multiply	M
urn:oasis:names:tc:xacml:1.0:function:integer-divide	M
urn:oasis:names:tc:xacml:1.0:function:double-divide	M
urn:oasis:names:tc:xacml:1.0:function:integer-mod	M
urn:oasis:names:tc:xacml:1.0:function:integer-abs	M
urn:oasis:names:tc:xacml:1.0:function:double-abs	M
urn:oasis:names:tc:xacml:1.0:function:round	M
urn:oasis:names:tc:xacml:1.0:function:floor	M
urn:oasis:names:tc:xacml:1.0:function:string-normalize-space	M
urn:oasis:names:tc:xacml:1.0:function:string-normalize-to-lower-case	M
urn:oasis:names:tc:xacml:1.0:function:double-to-integer	M
urn:oasis:names:tc:xacml:1.0:function:integer-to-double	M
urn:oasis:names:tc:xacml:1.0:function:or	M
urn:oasis:names:tc:xacml:1.0:function:and	M
urn:oasis:names:tc:xacml:1.0:function:n-of	M
urn:oasis:names:tc:xacml:1.0:function:not	M
urn:oasis:names:tc:xacml:1.0:function:integer-greater-than	M
urn:oasis:names:tc:xacml:1.0:function:integer-greater-than-or-equal	M

Функция	M/O
urn:oasis:names:tc:xacml:1.0:function:integer-less-than	M
urn:oasis:names:tc:xacml:1.0:function:integer-less-than-or-equal	M
urn:oasis:names:tc:xacml:1.0:function:double-greater-than	M
urn:oasis:names:tc:xacml:1.0:function:double-greater-than-or-equal	M
urn:oasis:names:tc:xacml:1.0:function:double-less-than	M
urn:oasis:names:tc:xacml:1.0:function:double-less-than-or-equal	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-add-dayTimeDuration	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-add-yearMonthDuration	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-subtract-dayTimeDuration	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-subtract-yearMonthDuration	M
urn:oasis:names:tc:xacml:1.0:function:date-add-yearMonthDuration	M
urn:oasis:names:tc:xacml:1.0:function:date-subtract-yearMonthDuration	M
urn:oasis:names:tc:xacml:1.0:function:string-greater-than	M
urn:oasis:names:tc:xacml:1.0:function:string-greater-than-or-equal	M
urn:oasis:names:tc:xacml:1.0:function:string-less-than	M
urn:oasis:names:tc:xacml:1.0:function:string-less-than-or-equal	M
urn:oasis:names:tc:xacml:1.0:function:time-greater-than	M
urn:oasis:names:tc:xacml:1.0:function:time-greater-than-or-equal	M
urn:oasis:names:tc:xacml:1.0:function:time-less-than	M
urn:oasis:names:tc:xacml:1.0:function:time-less-than-or-equal	M
urn:oasis:names:tc:xacml:2.0:function:time-in-range	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-greater-than	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-greater-than-or-equal	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-less-than	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-less-than-or-equal	M
urn:oasis:names:tc:xacml:1.0:function:date-greater-than	M
urn:oasis:names:tc:xacml:1.0:function:date-greater-than-or-equal	M
urn:oasis:names:tc:xacml:1.0:function:date-less-than	M
urn:oasis:names:tc:xacml:1.0:function:date-less-than-or-equal	M
urn:oasis:names:tc:xacml:1.0:function:string-one-and-only	M
urn:oasis:names:tc:xacml:1.0:function:string-bag-size	M
urn:oasis:names:tc:xacml:1.0:function:string-is-in	M
urn:oasis:names:tc:xacml:1.0:function:string-bag	M
urn:oasis:names:tc:xacml:1.0:function:boolean-one-and-only	M
urn:oasis:names:tc:xacml:1.0:function:boolean-bag-size	M
urn:oasis:names:tc:xacml:1.0:function:boolean-is-in	M
urn:oasis:names:tc:xacml:1.0:function:boolean-bag	M
urn:oasis:names:tc:xacml:1.0:function:integer-one-and-only	M
urn:oasis:names:tc:xacml:1.0:function:integer-bag-size	M
urn:oasis:names:tc:xacml:1.0:function:integer-is-in	M
urn:oasis:names:tc:xacml:1.0:function:integer-bag	M
urn:oasis:names:tc:xacml:1.0:function:double-one-and-only	M
urn:oasis:names:tc:xacml:1.0:function:double-bag-size	M
urn:oasis:names:tc:xacml:1.0:function:double-is-in	M
urn:oasis:names:tc:xacml:1.0:function:double-bag	M
urn:oasis:names:tc:xacml:1.0:function:time-one-and-only	M
urn:oasis:names:tc:xacml:1.0:function:time-bag-size	M
urn:oasis:names:tc:xacml:1.0:function:time-is-in	M
urn:oasis:names:tc:xacml:1.0:function:time-bag	M
urn:oasis:names:tc:xacml:1.0:function:date-one-and-only	M
urn:oasis:names:tc:xacml:1.0:function:date-bag-size	M
urn:oasis:names:tc:xacml:1.0:function:date-is-in	M

Функция	M/O
urn:oasis:names:tc:xacml:1.0:function:date-bag	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-one-and-only	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-bag-size	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-is-in	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-bag	M
urn:oasis:names:tc:xacml:1.0:function:anyURI-one-and-only	M
urn:oasis:names:tc:xacml:1.0:function:anyURI-bag-size	M
urn:oasis:names:tc:xacml:1.0:function:anyURI-is-in	M
urn:oasis:names:tc:xacml:1.0:function:anyURI-bag	M
urn:oasis:names:tc:xacml:1.0:function:hexBinary-one-and-only	M
urn:oasis:names:tc:xacml:1.0:function:hexBinary-bag-size	M
urn:oasis:names:tc:xacml:1.0:function:hexBinary-is-in	M
urn:oasis:names:tc:xacml:1.0:function:hexBinary-bag	M
urn:oasis:names:tc:xacml:1.0:function:base64Binary-one-and-only	M
urn:oasis:names:tc:xacml:1.0:function:base64Binary-bag-size	M
urn:oasis:names:tc:xacml:1.0:function:base64Binary-is-in	M
urn:oasis:names:tc:xacml:1.0:function:base64Binary-bag	M
urn:oasis:names:tc:xacml:1.0:function:dayTimeDuration-one-and-only	M
urn:oasis:names:tc:xacml:1.0:function:dayTimeDuration-bag-size	M
urn:oasis:names:tc:xacml:1.0:function:dayTimeDuration-is-in	M
urn:oasis:names:tc:xacml:1.0:function:dayTimeDuration-bag	M
urn:oasis:names:tc:xacml:1.0:function:yearMonthDuration-one-and-only	M
urn:oasis:names:tc:xacml:1.0:function:yearMonthDuration-bag-size	M
urn:oasis:names:tc:xacml:1.0:function:yearMonthDuration-is-in	M
urn:oasis:names:tc:xacml:1.0:function:yearMonthDuration-bag	M
urn:oasis:names:tc:xacml:1.0:function:x500Name-one-and-only	M
urn:oasis:names:tc:xacml:1.0:function:x500Name-bag-size	M
urn:oasis:names:tc:xacml:1.0:function:x500Name-is-in	M
urn:oasis:names:tc:xacml:1.0:function:x500Name-bag	M
urn:oasis:names:tc:xacml:1.0:function:rfc822Name-one-and-only	M
urn:oasis:names:tc:xacml:1.0:function:rfc822Name-bag-size	M
urn:oasis:names:tc:xacml:1.0:function:rfc822Name-is-in	M
urn:oasis:names:tc:xacml:1.0:function:rfc822Name-bag	M
urn:oasis:names:tc:xacml:2.0:function:string-concatenate	M
urn:oasis:names:tc:xacml:2.0:function:uri-string-concatenate	M
urn:oasis:names:tc:xacml:1.0:function:any-of	M
urn:oasis:names:tc:xacml:1.0:function:all-of	M
urn:oasis:names:tc:xacml:1.0:function:any-of-any	M
urn:oasis:names:tc:xacml:1.0:function:all-of-any	M
urn:oasis:names:tc:xacml:1.0:function:any-of-all	M
urn:oasis:names:tc:xacml:1.0:function:all-of-all	M
urn:oasis:names:tc:xacml:1.0:function:map	M
urn:oasis:names:tc:xacml:1.0:function:x500Name-match	M
urn:oasis:names:tc:xacml:1.0:function:rfc822Name-match	M
urn:oasis:names:tc:xacml:1.0:function:string-regexp-match	M
urn:oasis:names:tc:xacml:2.0:function:anyURI-regexp-match	M
urn:oasis:names:tc:xacml:2.0:function:ipAddress-regexp-match	M
urn:oasis:names:tc:xacml:2.0:function:dnsName-regexp-match	M
urn:oasis:names:tc:xacml:2.0:function:rfc822Name-regexp-match	M
urn:oasis:names:tc:xacml:2.0:function:x500Name-regexp-match	M
urn:oasis:names:tc:xacml:1.0:function:xpath-node-count	O
urn:oasis:names:tc:xacml:1.0:function:xpath-node-equal	O

Функция	M/O
urn:oasis:names:tc:xacml:1.0:function:xpath-node-match	O
urn:oasis:names:tc:xacml:1.0:function:string-intersection	M
urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of	M
urn:oasis:names:tc:xacml:1.0:function:string-union	M
urn:oasis:names:tc:xacml:1.0:function:string-subset	M
urn:oasis:names:tc:xacml:1.0:function:string-set-equals	M
urn:oasis:names:tc:xacml:1.0:function:boolean-intersection	M
urn:oasis:names:tc:xacml:1.0:function:boolean-at-least-one-member-of	M
urn:oasis:names:tc:xacml:1.0:function:boolean-union	M
urn:oasis:names:tc:xacml:1.0:function:boolean-subset	M
urn:oasis:names:tc:xacml:1.0:function:boolean-set-equals	M
urn:oasis:names:tc:xacml:1.0:function:integer-intersection	M
urn:oasis:names:tc:xacml:1.0:function:integer-at-least-one-member-of	M
urn:oasis:names:tc:xacml:1.0:function:integer-union	M
urn:oasis:names:tc:xacml:1.0:function:integer-subset	M
urn:oasis:names:tc:xacml:1.0:function:integer-set-equals	M
urn:oasis:names:tc:xacml:1.0:function:double-intersection	M
urn:oasis:names:tc:xacml:1.0:function:double-at-least-one-member-of	M
urn:oasis:names:tc:xacml:1.0:function:double-union	M
urn:oasis:names:tc:xacml:1.0:function:double-subset	M
urn:oasis:names:tc:xacml:1.0:function:double-set-equals	M
urn:oasis:names:tc:xacml:1.0:function:time-intersection	M
urn:oasis:names:tc:xacml:1.0:function:time-at-least-one-member-of	M
urn:oasis:names:tc:xacml:1.0:function:time-union	M
urn:oasis:names:tc:xacml:1.0:function:time-subset	M
urn:oasis:names:tc:xacml:1.0:function:time-set-equals	M
urn:oasis:names:tc:xacml:1.0:function:date-intersection	M
urn:oasis:names:tc:xacml:1.0:function:date-at-least-one-member-of	M
urn:oasis:names:tc:xacml:1.0:function:date-union	M
urn:oasis:names:tc:xacml:1.0:function:date-subset	M
urn:oasis:names:tc:xacml:1.0:function:date-set-equals	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-intersection	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-at-least-one-member-of	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-union	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-subset	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-set-equals	M
urn:oasis:names:tc:xacml:1.0:function:anyURI-intersection	M
urn:oasis:names:tc:xacml:1.0:function:anyURI-at-least-one-member-of	M
urn:oasis:names:tc:xacml:1.0:function:anyURI-union	M
urn:oasis:names:tc:xacml:1.0:function:anyURI-subset	M
urn:oasis:names:tc:xacml:1.0:function:anyURI-set-equals	M
urn:oasis:names:tc:xacml:1.0:function:hexBinary-intersection	M
urn:oasis:names:tc:xacml:1.0:function:hexBinary-at-least-one-member-of	M
urn:oasis:names:tc:xacml:1.0:function:hexBinary-union	M
urn:oasis:names:tc:xacml:1.0:function:hexBinary-subset	M
urn:oasis:names:tc:xacml:1.0:function:hexBinary-set-equals	M
urn:oasis:names:tc:xacml:1.0:function:base64Binary-intersection	M
urn:oasis:names:tc:xacml:1.0:function:base64Binary-at-least-one-member-of	M
urn:oasis:names:tc:xacml:1.0:function:base64Binary-union	M
urn:oasis:names:tc:xacml:1.0:function:base64Binary-subset	M
urn:oasis:names:tc:xacml:1.0:function:base64Binary-set-equals	M
urn:oasis:names:tc:xacml:1.0:function:dayTimeDuration-intersection	M

Функция	M/O
urn:oasis:names:tc:xacml:1.0:function:dayTimeDuration-at-least-one-member-of	M
urn:oasis:names:tc:xacml:1.0:function:dayTimeDuration-union	M
urn:oasis:names:tc:xacml:1.0:function:dayTimeDuration-subset	M
urn:oasis:names:tc:xacml:1.0:function:dayTimeDuration-set-equals	M
urn:oasis:names:tc:xacml:1.0:function:yearMonthDuration-intersection	M
urn:oasis:names:tc:xacml:1.0:function:yearMonthDuration-at-least-one-member-of	M
urn:oasis:names:tc:xacml:1.0:function:yearMonthDuration-union	M
urn:oasis:names:tc:xacml:1.0:function:yearMonthDuration-subset	M
urn:oasis:names:tc:xacml:1.0:function:yearMonthDuration-set-equals	M
urn:oasis:names:tc:xacml:1.0:function:x500Name-intersection	M
urn:oasis:names:tc:xacml:1.0:function:x500Name-at-least-one-member-of	M
urn:oasis:names:tc:xacml:1.0:function:x500Name-union	M
urn:oasis:names:tc:xacml:1.0:function:x500Name-subset	M
urn:oasis:names:tc:xacml:1.0:function:x500Name-set-equals	M
urn:oasis:names:tc:xacml:1.0:function:rfc822Name-intersection	M
urn:oasis:names:tc:xacml:1.0:function:rfc822Name-at-least-one-member-of	M
urn:oasis:names:tc:xacml:1.0:function:rfc822Name-union	M
urn:oasis:names:tc:xacml:1.0:function:rfc822Name-subset	M
urn:oasis:names:tc:xacml:1.0:function:rfc822Name-set-equals	M

8 Профиль основного и иерархического контроля доступа на ролевой основе (RBAC)

В этом пункте определяется профиль для использования XACML для обеспечения потребностей "основного" и "иерархического" контроля доступа на ролевой основе (RBAC).

8.1 Базовая информация о RBAC

Этот пункт является информативным.

В этом пункте определен профиль для использования с языком XACML, чтобы соответствовать требованиям "основного" и "иерархического" контроля доступа на ролевой основе (RBAC).

ПРИМЕЧАНИЕ. – Для получения информации о RBAC смотрите [RBAC].

8.1.1 Область применения

При контроле доступа на ролевой основе позволено устанавливать стратегии в терминах ролей субъекта, а не строго в терминах идентичностей индивидуальных субъектов. Это важно для масштабируемости и управляемости систем контроля доступом.

Стратегии, установленные в этом профиле, могут отвечать на три типа вопросов:

1) Если у субъекта имеются задействованные роли R_1, R_2, ... R_n, может ли субъект X получить доступ к заданному ресурсу, используя заданное действие?
2) Позволено ли субъекту X иметь задействованную роль R_i?
3) Если у субъекта имеются задействованные роли R_1, R_2, ... R_n, означает ли это, что субъекту будут предоставлены разрешения, связанные с заданной ролью R'? То есть, является ли роль R' равной или *junior* по отношению к любой роли R_1, R_2, ... R_n?

Стратегии, установленные в этом профиле, не отвечают на вопрос "Какой набор ролей имеется у субъекта X?" Этот вопрос должен обрабатываться органом задействования ролей, а не напрямую пунктом PDP XACML. Такой объект может воспользоваться стратегиями XACML, но ему понадобится дополнительная информация.

Стратегии, установленные в этом профиле, допускают все роли для заданного субъекта, которые уже были задействованы в момент запроса решения об авторизации. Они не рассматривают среду, в которой роли должны задействоваться динамично, основываясь на ресурсе или действиях, которые пытается выполнить субъект. По этой причине стратегии, установленные в этом профиле, также не рассматривают статическое или динамическое "разделение обязанностей". Будущий профиль может обращаться к требованиям этого типа среды.

8.1.2 Роль

В данной Рекомендации роли выражены в виде атрибутов субъекта XACML. Есть два исключения: в Назначении роли `<PolicySet>` или `<Policy>` и в `HasPrivilegesOfRole` `<Policy>`, роль появляется в виде атрибута ресурса.

Ролевые атрибуты могут быть выражены двумя способами, в зависимости от требований прикладной среды. В некоторых средах может быть небольшое количество "ролевых атрибутов", в которых имя каждого такого атрибута это какое-то имя, показывающее "роль", и в которых значение каждого такого атрибута показывает имя исполняемой роли. Например, в этом первом типе среды может быть один "ролевой атрибут", у которого есть AttributeId "&role;" (этот профиль рекомендует использовать такой идентификатор). Возможными ролями являются значения этого одного атрибута, и это могут быть "&roles;officer", "&roles;manager", и "&roles;employee". Такой способ выражения ролей лучше всего действует со стратегиями, выраженными языком XACML. Этот метод идентификации ролей является также самым подходящим для взаимодействия.

С другой стороны, в других прикладных средах может быть несколько разных идентификаторов атрибутов, каждый из которых показывает другую роль. Например, во втором типе среды может быть три идентификатора атрибутов: "urn:someapp:attributes:officer-role", "urn:someapp:attributes:manager-role", и "urn:someapp:attributes:employee-role". В таком случае значением атрибута может быть "пустой" или он может содержать различные параметры, связанные с данной ролью. Стратегии XACML могут обрабатывать роли, выраженные таким способом, но не так легко, как роли, выраженные первым способом.

Язык XACML поддерживает множество субъектов в каждом запросе о доступе, показывая различные объекты, которые могут участвовать в создании запроса. Например, обычно есть человек-пользователь, который инициирует запрос, хотя бы и косвенным образом. Существует обычно одно или более приложений или оснований кода, которые создают фактический запрос о доступе низкого уровня от лица пользователя. Имеется вычислительное устройство, на котором выполняется это приложение или основание кода. И у этого устройства может быть идентичность, такая как адрес IP. В XACML идентифицируется каждый такой Subject с помощью атрибута xml SubjectCategory, который показывает тип описываемого субъекта. Например, у человека-пользователя есть SubjectCategory с &subject-category;access-subject (это категория по умолчанию); у приложения, которое создает запрос о доступе, есть SubjectCategory с &subject-category;codebase и т. д. В этом профиле ролевой атрибут может быть связан с любой категорией субъектов, участвующих в создании запроса о доступе.

8.1.3 Стратегии

В данной Рекомендации установлено четыре типа стратегий.

1) <PolicySet> **Роли** или **RPS**: <PolicySet>, который связывает держателей заданного ролевого атрибута и значения с <PolicySet> Разрешения, в котором содержатся фактические разрешения, связанные с заданной ролью. Элемент <Target> типа стратегии <PolicySet> Роли ограничивает применяемость <PolicySet> к субъектам, держащим связанный ролевой атрибут и значение. Каждый <PolicySet> Роли ссылается на соответствующий <PolicySet> Разрешения, но не содержит и не ссылается на любой другой элемент <Policy> или <PolicySet>.

2) <PolicySet> **Разрешения** или **PPS**: <PolicySet>, в котором содержатся фактические разрешения, связанные с заданной ролью. В нем содержатся элементы <Policy> и <Rules>, которые описывают ресурсы и действия, к которым разрешен доступ для субъектов, наряду с дальнейшими условиями по этому доступу, такими как время дня. В заданном <PolicySet> Разрешения могут также содержаться ссылки на типы <PolicySet> Разрешения, связанные с другими ролями, которые являются младшими по отношению к заданной роли, таким образом позволяя заданному <PolicySet> Разрешения унаследовать все разрешения, связанные с ролью того типа стратегии Permission <PolicySet>, на который дана ссылка. Элемент <Target> типа стратегии Permission <PolicySet>, если он присутствует, не должен ограничивать субъекты, к которым применим этот <PolicySet>.

3) <Policy> или <PolicySet> **Назначения ролей**: <Policy> или <PolicySet>, который определяет, какие роли могут быть задействованы или назначены каким субъектам. Он также может устанавливать ограничения на комбинации ролей или на общее количество ролей, которые могут быть назначены или задействованы для заданного субъекта. Этот тип стратегии используется органом задействования ролей. Использование Назначения ролей <Policy> или <PolicySet> является необязательным.

4) **HasPrivilegesOfRole** <Policy>: <Policy> в <PolicySet> Разрешения, который поддерживает запросы, спрашивающие есть ли у субъекта привилегии, связанные с этой ролью. Если этот тип запроса должен поддерживаться, HasPrivilegesOfRole <Policy> должен быть включен в каждый <PolicySet> Разрешения. Поддержка этого типа <Policy> и, таким образом, для запросов, спрашивающих есть ли у субъекта привилегии, связанные с этой ролью, является необязательной.

Экземпляры <PolicySet> Разрешения должны храниться в репозитории стратегии таким образом, чтобы их никогда нельзя было использовать в качестве исходной стратегии для пункта PDP XACML; экземпляры <PolicySet> Разрешения должны быть досягаемы только через соответствующую <PolicySet> Роли. Это сделано для того, чтобы, с целью поддержания иерархических ролей, <PolicySet> Разрешения применялся к каждому субъекту. <PolicySet> Разрешения зависит от его соответствующего <PolicySet> Роли, чтобы гарантировать тот факт, что только субъекты, держащие соответствующие ролевые атрибуты, получат доступ к разрешениям в заданном <PolicySet> Разрешения.

Использование отдельных экземпляров <PolicySet> Роли и <PolicySet> Разрешения позволяет поддерживать иерархические RBAC, в которых более *senior* роль может получить разрешения более *junior* роли. `<PolicySet>` Разрешения, который не ссылается на другие элементы `<PolicySet>` Разрешения, может фактически быть `<Policy>`, а не `<PolicySet>` языка XACML. Требование о том, чтобы это был `<PolicySet>`, однако, позволяет связанной с ним роли стать частью ролевой иерархии позже, не требуя никаких изменений для других стратегий.

8.1.4 Многоролевые разрешения

В этом профиле есть возможность выразить стратегии, в которых пользователь должен держать несколько ролей одновременно, для того чтобы получить доступ к нескольким разрешениям. Например, при изменении инструкций по уходу за пациентом больницы может потребоваться, чтобы у `Subject`, выполняющего действие, имелась роль как врача, так и роль персонала.

Эти стратегии могут быть выражены с использованием `<PolicySet>` Роли, в которой элемент `<Target>` требует, чтобы у `Subject` имелись все необходимые ролевые атрибуты. Это требование выполняется с помощью использования отдельного элемента `<Subject>`, содержащего множество элементов `<SubjectMatch>`. Связанный тип <PolicySet> Разрешения должен устанавливать разрешения, связанные с `Subjects`, у которых одновременно имеются все установленные роли в задействованном состоянии.

`<PolicySet>` Разрешения, связанный с многоролевой стратегией, может ссылаться на экземпляры `<PolicySet>` Разрешения, связанные с другими ролями и, таким образом, может унаследовать разрешения от других ролей. Разрешения, связанные с заданным многоролевым `<PolicySet>` могут также быть унаследованными другой ролью, если в другую роль включена ссылка на `<PolicySet>` Разрешения, связанный с многоролевой стратегией в своей собственной `<PolicySet>`.

8.2 Пример RBAC

Этот пункт является информативным.

В этом пункте представлен полный пример типов стратегий, связанных с контролем доступа на ролевой основе.

В управлении организацией используется две роли: управляющего и служащего. В этом примере они выражены двумя отдельными значениями для единого атрибута XACML с `AttributeId` "`&role;`". Значениями атрибута `&role;`, относящиеся к двум ролям, являются: "`&roles;employee`" и "`&roles;manager`". У служащего есть разрешение на создание заказа на покупку. У управляющего есть разрешение на подпись заказа на покупку плюс любое разрешение, связанное с ролью служащего. Таким образом, роль управляющего является старшей, относительно роли служащего, а роль служащего является младшей, относительно роли управляющего.

В соответствии с этим профилем будет два экземпляра `<PolicySet>` Разрешения: один для роли управляющего и один для роли служащего. <PolicySet> Разрешения управляющего даст любому `Subject` особое разрешение подписывать заказ на покупку и будет ссылаться на <PolicySet> Разрешения служащего, для того чтобы унаследовать его разрешения. <PolicySet> Разрешения служащего даст любому `Subject` разрешение на создание заказа на покупку.

В соответствии с этим профилем будет также два экземпляра `<PolicySet>` Роли: один для роли управляющего и один для роли служащего. В <PolicySet> Роли управляющего будет содержаться `<Target>`, требующий, чтобы `Subject` держал `&role;` атрибут со значением "`&roles;manager`". Он будет ссылаться на <PolicySet> Разрешения управляющего. В <PolicySet> Роли служащего будет содержаться `<Target>`, требующий, чтобы `Subject` держал `&role;` атрибут со значением "`&roles;employee`". Он будет ссылаться на <PolicySet> Разрешения служащего.

8.2.1 <PolicySet> Разрешения для роли управляющего

В следующем <PolicySet> Разрешения содержатся разрешения, связанные с ролью управляющего. Стратегия поиска пункта PDP должна быть организована таким образом, чтобы доступ к этому `<PolicySet>` предоставлялся только по ссылке от <PolicySet> Роли управляющего (смотрите таблицу 8-1).

Таблица 8-1/X.1142 – <PolicySet> Разрешения для управляющих

```
<PolicySet xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
PolicySetId="PPS:manager:role"
PolicyCombiningAlgId="&policy-combine;permit-overrides">

<!-- Permissions specifically for the manager role -->
<Policy PolicyId="Permissions:specifically:for:the:manager:role"
    RuleCombiningAlgId="&rule-combine;permit-overrides">
    <!-- Permission to sign a purchase order -->
    <Rule RuleId="Permission:to:sign:a:purchase:order" Effect="Permit">
                <Target>
                      <Resources>
                            <Resource>
                                  <ResourceMatch MatchId="&function;string-equal">
                                     <AttributeValue
                                          DataType="&xml;string">purchase
order</AttributeValue>
                                  <ResourceAttributeDesignator
AttributeId="&resource;resource-id"
                                          DataType="&xml;string"/>
                            </ResourceMatch>
                            </Resource>
                      </Resources>
                      <Actions>
                            <Action>
                                  <ActionMatch MatchId="&function;string-equal">
                                    <AttributeValue
                                       DataType="&xml;string">sign</AttributeValue>
                                       ActionAttributeDesignator
                                       AttributeId="&action;action-id"
                                       DataType="&xml;string"/>
                                  </ActionMatch>
                            </Action>
                      </Actions>
                </Target>
          </Rule>
</Policy>
<!-- Include permissions associated with employee role -->
<PolicySetIdReference>PPS:employee:role</PolicySetIdReference>
</PolicySet>
```

8.2.2 <PolicySet> Разрешения для роли служащего

В следующем <PolicySet> Разрешения содержатся разрешения, связанные с ролью служащего (смотрите таблицу 8-2). Стратегия поиска пункта PDP должна быть организована таким образом, чтобы доступ к этому `<PolicySet>` предоставлялся только по ссылке от <PolicySet> Роли служащего или по ссылке от <PolicySet> Роли управляющего более высокого ранга через <PolicySet> Разрешения управляющего.

Таблица 8-2/X.1142 – <PolicySet> Разрешения для служащих

```
<PolicySet xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
   PolicySetId="PPS:employee:role"
   PolicyCombiningAlgId="&policy-combine;permit-overrides">
   <!-- Permissions specifically for the employee role -->
   <Policy PolicyId="Permissions:specifically:for:the:employee:role"
         RuleCombiningAlgId="&rule-combine;permit-overrides">
   <!-- Permission to create a purchase order -->
         <Rule RuleId="Permission:to:create:a:purchase:order" Effect="Permit">
               <Target>
                     <Resources>
                           <Resource>
                                 <ResourceMatch MatchId="&function;string-equal">
                                   <AttributeValue
                                   DataType="&xml;string">purchase
order</AttributeValue>
                                   <ResourceAttributeDesignator
AttributeId="&resource;resource-id"
                                   DataType="&xml;string"/>
                                 </ResourceMatch>
                           </Resource>
                     </Resources>
                     <Actions>
                           <Action>
                                 <ActionMatch MatchId="&function;string-equal">
                                   <AttributeValue
DataType="&xml;string">create</AttributeValue>
                                   <ActionAttributeDesignator
AttributeId="&action;action-id"
                                   DataType="&xml;string"/>
                                 </ActionMatch>
                           </Action>
                     </Actions>
               </Target>
         </Rule>
   </Policy>
</PolicySet>
```

8.2.3 <PolicySet> Роли для роли управляющего

Следующий <PolicySet> Роли применим, в соответствии с его `<Target>`, только к `Subjects`, которые держат `&role;` (смотрите таблицу 8-3); атрибут со значением "`&roles;manager`". `<PolicySetIdReference>` указывает на `<PolicySet>` Разрешения, связанный с ролью управляющего.

Таблица 8-3/X.1142 – <PolicySet> Роли для управляющих

```
<PolicySet xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
   PolicySetId="RPS:manager:role"
   PolicyCombiningAlgId="&policy-combine;permit-overrides">
   <Target>
         <Subjects>
               <Subject>
                     <SubjectMatch MatchId="&function;anyURI-equal">
                           <AttributeValue
DataType="&xml;anyURI">&roles;manager</AttributeValue>
                           <SubjectAttributeDesignator AttributeId="&role;"
                                 DataType="&xml;anyURI"/>
                     </SubjectMatch>
               </Subject>
         </Subjects>
   </Target>
<!-- Use permissions associated with the manager role -->
   <PolicySetIdReference>PPS:manager:role</PolicySetIdReference>
</PolicySet>
```

8.2.4 <PolicySet> Роли для роли служащего

Следующий <PolicySet> Роли применим, в соответствии с его `<Target>`, только к `Subjects`, которые держат `&role;` (смотрите таблицу 8-4); атрибут со значением "`&roles;employee`". `<PolicySetIdReference>` указывает на `<PolicySet>` Разрешения, связанный с ролью служащего.

Таблица 8-4/X.1142 – <PolicySet> Роли для служащих

```
<PolicySet xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
   PolicySetId="RPS:employee:role"
   PolicyCombiningAlgId="&policy-combine;permit-overrides">
   <Target>
          <Subjects>
                 <Subject>
                        <SubjectMatch MatchId="&function;anyURI-equal">
                               <AttributeValue
DataType="&xml;anyURI">&roles;employee</AttributeValue>
                               <SubjectAttributeDesignator AttributeId="&role;"
DataType="&xml;anyURI"/>
                        </SubjectMatch>
                 </Subject>
          </Subjects>
   </Target>
<!-- Use permissions associated with the employee role -->
<PolicySetIdReference>PPS:employee:role</PolicySetIdReference>
</PolicySet>
```

8.2.5 HasPrivilegesOfRole стратегии и запросы

Система RBAC XACML может поддерживать запросы по форме "У этого субъекта есть привилегии роли X?" Если это так, то каждый <PolicySet> Разрешения должен содержать `HasPrivilegesOfRole <Policy>`. Для <PolicySet> Разрешения для управляющих `HasPrivilegesOfRole <Policy>` выглядел бы, как в таблице 8-5.

Таблица 8-5/X.1142 – <PolicySet> Разрешения для управляющих

```
<!-- HasPrivilegesOfRole Policy for manager role -->
<Policy PolicyId="Permission:to:have:manager:role:permissions"
   RuleCombiningAlgId="&rule-combine;permit-overrides">
<!-- Permission to have manager role permissions -->
   <Rule RuleId="Permission:to:have:manager:permissions" Effect="Permit">
          <Condition>
                 <Apply FunctionId="&function;and">
                        <Apply FunctionId="&function;anyURI-is-in">
                               <AttributeValue
DataType="&xml;anyURI">&roles;manager</AttributeValue>
                               <ResourceAttributeDesignator AttributeId="&role;"
DataType="&xml;anyURI"/>
                        </Apply>
                               <Apply FunctionId="&function;anyURI-is-in">
                               <AttributeValue

   DataType="&xml;anyURI">&actions;hasPrivilegesofRole</AttributeValue>
                               <ActionAttributeDesignator AttributeId="&action;action-
id"
                                      DataType="&xml;anyURI"/>
                        </Apply>
                 </Apply>
          </Condition>
   </Rule>
</Policy>
```

Для <PolicySet> Разрешения для служащих `HasPrivilegesOfRole <Policy>` выглядел бы, как в таблице 8-6.

Таблица 8-6/X.1142 – <PolicySet> Разрешения для служащих

```
<!-- HasPrivilegesOfRole Policy for employee role -->
<Policy PolicyId="Permission:to:have:employee:role:permissions"
   RuleCombiningAlgId="&rule-combine;permit-overrides">
<!-- Permission to have employee role permissions -->
   <Rule RuleId="Permission:to:have:employee:permissions" Effect="Permit">
         <Condition>
               <Apply FunctionId="&function;and">
                     <Apply FunctionId="&function;anyURI-is-in">
                           <AttributeValue
DataType="&xml;anyURI">&roles;employee</AttributeValue>
                           <ResourceAttributeDesignator AttributeId="&role;"
                                 DataType="&xml;anyURI"/>
                     </Apply>
                     <Apply FunctionId="&function;anyURI-is-in">
                     <AttributeValue
DataType="&xml;anyURI">&actions;hasPrivilegesofRole
                     </AttributeValue>
                     <ActionAttributeDesignator AttributeId="&action;action-id"
                           DataType="&xml;anyURI"/>
                     </Apply>
               </Apply>
         </Condition>
   </Rule>
</Policy>
```

Запрос, спрашивающий, есть ли у субъекта `Anne` привилегии, связанные с `&roles;manager`, выглядел бы, как в таблице 8-7.

Таблица 8-7/X.1142 – Запрос о субъекте

```
<Request>
   <Subject>
         <Attribute AttributeId="&subject;subject-id" DataType="&xml;string">
               <AttributeValue>Anne</AttributeValue>
         </Attribute>
   </Subject>
   <Resource>
         <Attribute AttributeId="&role;" DataType="&xml;anyURI">
               <AttributeValue>&roles;manager</AttributeValue>
               </Attribute>
   </Resource>
   <Action>
         <Attribute AttributeId="&action;action-id"

   DataType="&xml;anyURI">&actions;hasPrivilegesOfRole</AttributeValue>
         </Attribute>
   </Action>
</Request>
```

Либо в `<Request>` должны содержаться прямые роли `Anne` (в этом случае, `&roles;employee`), либо обработчик контекста PDP должен быть в состоянии их обнаружить. Стратегии `HasPrivilegesOfRole` не выполняют работу по связыванию ролей с субъектами.

8.3 Назначение и задействование ролевых атрибутов

Этот пункт является информативным.

Назначение различных ролевых атрибутов для пользователей и задействование этих атрибутов внутри сеанса выходит за рамки данного PDP XACML. Должен существовать один или более отдельных объектов, относящихся к органам задействования ролей, реализованных для выполнения этих функций. В этом профиле предполагается, что присутствие в контексте запроса XACML ролевого атрибута для заданного пользователя (`Subject`), является действительным назначением в момент запроса решения о доступе.

Итак, откуда поступают ролевые атрибуты для субъекта? Как выглядит один из этих органов задействования ролей? Ответ зависит от реализации, но можно предположить некоторые возможности.

В некоторых случаях ролевые атрибуты могут поступить из службы управления идентичностью, в которой хранится информация о пользователе, в том числе назначенные или разрешенные роли для субъекта; служба управления идентичностью действует, как орган задействования ролей. Эта служба может хранить статичные

ролевые атрибуты в справочнике LDAP, а обработчик контекста PDP может отыскивать их там. Или эта служба может отвечать на запросы о ролевых атрибутах для субъектов, поступающих от обработчика контекста PDP, в которых запросы находятся в форме запросов об атрибутах SAML.

Органы задействования атрибутов могут использовать <Policy> или <PolicySet> присвоение ролей XACML для выяснения того обстоятельства, есть ли у субъекта разрешение на задействование конкретного ролевого атрибута и значения. <Policy> или <PolicySet> Назначения ролей отвечает на вопрос "Позволено ли субъекту X иметь задействованную роль Ri?". Она не отвечает на вопрос "Какой набор ролей разрешено задействовать субъекту X?" У органа задействования ролей должен быть какой-то способ узнать, на какую роль или роли делать заявку. Например, в органе задействования ролей может храниться список всех возможных ролей и, когда нужно узнать о ролях, связанных с заданным субъектом, надо сделать запрос о стратегиях назначения ролей для каждой кандидатуры роли.

В этом профиле стратегии назначения ролей отличаются от набора экземпляров <PolicySet> Роли и `<PolicySet>` Разрешения, используемых для выяснения разрешений доступа, связанных с каждой ролью. Стратегии Назначения ролей должны использоваться, только если запрос XACML поступает от органа задействования ролей. Этим разделением можно управлять различными способами, например, используя разные пункты PDP с разными запасами стратегий или запрашивая элементы `<Request>` для запросов о задействовании ролей, чтобы включить `<Subject>` с `SubjectCategory` со значением "`&subject-category;role-enablement-authority`".

Не существует фиксированного формата для `<Policy>` Назначения ролей. В следующем примере (таблица 8-8) иллюстрируется один из возможных форматов. В нем содержатся два элемента `<Rule>` XACML. В первом `<Rule>` утверждается, что `Anne` и `Seth`, и `Yassir` разрешено иметь роль "`&roles;employee`", задействованную между 9 утра и 5 вечера. Во втором `<Rule>` утверждается, что `Steve` разрешено иметь роль "`&roles;manager`", задействованную без ограничений по времени дня.

Таблица 8-8/X.1142 – Пример назначения ролей

```
<Policy xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
    PolicyId="Role:Assignment:Policy"
    RuleCombiningAlgId="&rule-combine;permit-overrides">
<!-- Employee role requirements rule -->
    <Rule RuleId="employee:role:requirements" Effect="Permit">
        <Target>
            <Subjects>
                <Subject>
                    <SubjectMatch MatchId="&function;string-equal">
                        <AttributeValue DataType="&xml;string">Seth</AttributeValue>
                        <SubjectAttributeDesignator AttributeId="&subject;subject-id"
                            DataType="&xml;string"/>
                    </SubjectMatch>
                </Subject>
                <Subject>
                    <SubjectMatch MatchId="&function;string-equal">
                        <AttributeValue DataType="&xml;string">Anne</AttributeValue>
                        <SubjectAttributeDesignator AttributeId="&subject;subject-id"
                            DataType="&xml;string"/>
                    </SubjectMatch>
                </Subject>
            </Subjects>
        <Resources>
            <Resource>
                <ResourceMatch MatchId="&function;anyURI-equal">
                    <AttributeValue
DataType="&xml;anyURI">&roles;employee</AttributeValue>
                        <ResourceAttributeDesignator AttributeId="&role;"
                            DataType="&xml;anyURI"/>
                    </ResourceMatch>
                    </Resource>
                </Resources>
                <Actions>
                <Action>
                    <ActionMatch MatchId="&function;anyURI-equal">
                        <AttributeValue DataType="&xml;anyURI">&actions;
                            enableRole</AttributeValue>
                        <ActionAttributeDesignator AttributeId="&action;action-id"
                            DataType="&xml;anyURI"/>
                    </ActionMatch>
```

Таблица 8-8/X.1142 – Пример назначения ролей

```
                </Action>
            </Actions>
        </Target>
        <Condition>
            <Apply FunctionId="&function;and">
                <Apply FunctionId="&function;time-greater-than-or-equal">
                    <Apply FunctionId="&function;time-one-and-only">
                        <EnvironmentAttributeDesignator
AttributeId="&environment;current-time"
                            DataType="&xml;time"/>
                    </Apply>
                    <AttributeValue DataType="&xml;time">9h</AttributeValue>
                </Apply>
                <Apply FunctionId="&function;time-less-than-or-equal">
                    <Apply FunctionId="&function;time-one-and-only">
                        <EnvironmentAttributeDesignator
                             AttributeId="&environment;current-time"
DataType="&xml;time"/>
                    </Apply>
                    <AttributeValueDataType="&xml;time">17h</AttributeValue>
                </Apply>
        </Apply>
    </Condition>
</Rule>
<!-- Manager role requirements rule -->
    <Rule RuleId="manager:role:requirements" Effect="Permit">
        <Target>
            <Subjects>
                <Subject>
                    <SubjectMatch MatchId="&function;string-equal">
                        <AttributeValue DataType="&xml;string">Steve</AttributeValue>
                        <SubjectAttributeDesignator AttributeId="&subject;subject-id"
                            DataType="&xml;string"/>
                    </SubjectMatch>
                </Subject>
            </Subjects>
            <Resources>
                <Resource>
                    <ResourceMatch MatchId="&function;anyURI-equal">
                        <AttributeValue
                            DataType="&xml;anyURI">&roles;:manager</AttributeValue>
                        <ResourceAttributeDesignator AttributeId="&role;"
                            DataType="&xml;anyURI"/>
                    </ResourceMatch>
                </Resource>
            </Resources>
        <Actions>
            <Action>
                <ActionMatch MatchId="&function;anyURI-equal">
                    <AttributeValue
                        DataType="&xml;anyURI">&actions;enableRole</AttributeValue>
                        <ActionAttributeDesignator AttributeId="&action;action-id"
                            DataType="&xml;anyURI"/>
                </ActionMatch>
            </Action>
        </Actions>
    </Target>
</Rule>
</Policy>
```

8.4 Реализация модели RBAC

Этот пункт является информативным.

В следующих пунктах описывается, как использовать стратегии XACML для реализации различных компонентов модели RBAC (смотрите [RBAC]).

8.4.1 Основной RBAC

В основной RBAC включены следующие пять базовых элементов данных:

- Пользователей реализуют, используя `Subjects` XACML. Могут использоваться любые подходящие значения `SubjectCategory` XACML `SubjectCategory`.
- Роли выражают, используя один или более атрибутов субъекта XACML. Набор ролей в большой степени зависит от приложения и домена стратегии, и очень важно, чтобы не смешивалось разное использование ролей. По этой причине этот профиль не стремится описать какой-то стандартный набор ролевых значений, хотя в этом профиле все же рекомендуется использовать распространенное значение `AttributeId`: "`urn:oasis:names:tc:xacml:2.0:subject:role`". Рекомендуется, чтобы в каждом приложении или домене стратегии был согласован и оглашен уникальный набор значений `AttributeId`, значений `DataType`, и значений `<AttributeValue>`, которые будут использоваться для различных ролей, свойственных этому домену.
- Объекты выражают, используя `Resources` XACML.
- Операции выражают, используя `Actions` XACML.
- Разрешения выражают, используя экземпляры <PolicySet> Роли и <PolicySet> Разрешения XACML, как описано в предыдущих пунктах.

В основном RBAC требуется поддержка для множества пользователей на каждую роль, множества ролей на каждого пользователя, множества разрешений на каждую роль и множества ролей на каждое разрешение. Каждое из этих требований может быть выполнено с помощью стратегий XACML, основанных на этом профиле, следующим образом. Заметим однако, что фактическое Назначения ролей для пользователей выходит за рамки данного PDP XACML.

В языке XACML допускается, чтобы множество субъектов было связано с заданным ролевым атрибутом. Стратегии <PolicySet> Роли XACML, определенные в терминах владения конкретным ролевым `<Attribute>` и `<AttributeValue>`, будут применимы к любому запрашивающему пользователю, для которого этот ролевой `<Attribute>` и `<AttributeValue>` находятся в контексте запроса XACML.

В языке XACML допускается, чтобы множество ролевых атрибутов или значений ролевых атрибутов было связано с заданным `Subject`. Если у `Subject` имеется множество задействованных ролей, то любой экземпляр `<PolicySet>` Роли, примененный к любой из этих ролей, может быть оценен и разрешения в соответствующем <PolicySet> Разрешения будут получены. Даже возможно определить стратегии, которым требуется, чтобы у заданного `Subject` было множество задействованных ролевых атрибутов или значений одновременно. В этом случае разрешения, связанные с требованием множества ролей, будут применимы только к `Subject`, у которого есть все необходимые ролевые атрибуты и значения в тот момент, когда контекст запроса XACML представляется в пункт PDP для оценки.

`<PolicySet>` Разрешения, связанный с заданной ролью, может разрешить доступ к множеству ресурсов, используя множество действий. У XACML имеется богатый набор структурных компонентов для составления разрешений, поэтому существует множество способов, с помощью которых могут быть выражены роли для множества разрешений. Любая Роль A может быть связана с <PolicySet> Разрешения B с помощью включения `<PolicySetIdReference>` в <PolicySet> Разрешения B, находящийся в `<PolicySet>` Разрешения, связанный с Ролью A. Таким образом, тот же самый набор разрешений может быть связан с более, чем одной ролью.

Вдобавок к базовым требованиям к основному RBAC стратегии XACML, использующие этот профиль, могут также выразить произвольные условия для применения конкретных разрешений, связанных с этой ролью. В эти условия могут входить ограничения на выдачу разрешения в заданный период времени в течение дня или ограничения на выдачу разрешений для держателей роли, которые также владеют каким-то другим атрибутом, будь то ролевой атрибут или нет.

8.4.2 Иерархический RBAC

Иерархический RBAC расширяет основной RBAC с помощью возможности определения наследуемых связей между ролями. Например, роль A может быть определена, как наследующая все разрешения, связанные с ролью B. В этом случае, роль A считается старшей по отношению к роли B в ролевой иерархии. Если роль B, в свою очередь, наследует разрешения, связанные с ролью C, то роль A будет также наследовать все те разрешения, на основании того, что она является старшей по отношению к роли B.

Стратегии XACML, использующие этот профиль, могут реализовывать ролевую иерархию, вводя `<PolicySetIdReference>` в `<PolicySet>` Разрешения, связанный с одной ролью, внутрь `<PolicySet>` Разрешения, связанного с другой ролью. Та роль, в которую включена `<PolicySetIdReference>`, будет затем наследовать разрешения, связанные с упомянутой ролью.

Этот профиль структурирует стратегии таким образом, что наследуемые свойства могут быть добавлены к роли в любое время, без требования каких-либо изменений в экземплярах `<PolicySet>`, связанных с любыми другими ролями. Организация первоначально может не использовать ролевую иерархию, но может позже извлечь пользу из этой функциональной возможности, без необходимости переписывания существующих стратегий.

8.5 Профиль

В этом пункте обсуждаются роли, ролевые атрибуты, Назначения ролей и контроль доступа.

8.5.1 Роли и ролевые атрибуты

Роли должны выражаться с использованием одного или более атрибутов XACML. В каждом прикладном домене, использующем этот профиль для контроля доступа на ролевой основе, должны быть определены и согласованы одно или более значений AttributeId, чтобы использовать их в ролевых атрибутах. Каждое такое значение атрибута AttributeId должно быть связано с набором разрешенных значений и их DataTypes. У каждого разрешенного значения для такого AttributeId должна быть четко определенная семантика для использования соответствующих значений в стратегиях.

В этом профиле рекомендуется использовать значение AttributeId: "urn:oasis:names:tc:xacml:2.0:subject:role" для всех ролевых атрибутов. У экземпляров этого атрибута должен быть DataType со значением "http://www.w3.org/2001/XMLSchema#anyURI".

8.5.2 Назначение или задействование ролей

Орган задействования ролей, ответственный за Назначение ролей пользователям и за задействование ролей для их использования в рамках сеанса пользователя, может использовать <Policy> или <PolicySet> Назначение ролей XACML для выяснения, каким пользователям разрешено задействовать какие роли и при каких условиях. Не существует установленного формата для <Policy> или <PolicySet> Назначения ролей. Рекомендуется, чтобы роли в <Policy> или <PolicySet> Назначения ролей выражались в атрибутах, в которых AttributeId является &role; а <AttributeValue> является URI для подходящих ролевых значений. Рекомендуется, чтобы действие назначения или задействования роли выражалось, как атрибут действия, в котором AttributeId является &action;action-id, DataType является &xml;anyURI, и <AttributeValue> является &actions;enableRole.

8.5.3 Контроль доступа

Контроль доступа на ролевой основе должен быть реализован с использованием двух типов <PolicySet>: <PolicySet> Роли, <PolicySet> Разрешения. Конкретные функции и требования этих двух типов <PolicySet> следующие.

Для каждой роли должен быть определена один <PolicySet> Роли. В таком <PolicySet> должен содержаться элемент <Target>, который ограничивает применение <PolicySet> только теми субъектами, у которых имеется атрибут XACML, связанный с заданной ролью; элемент <Target> не должен ограничивать Resource, Action или Environment. В каждом <PolicySet> Роли должен содержаться отдельный элемент <PolicySetIdReference>, который ссылается на уникальный Permission <PolicySet>,Разрешение связанный с этой ролью. В <PolicySet> Роли не должны содержаться любые другие элементы <Policy>, <PolicySet>, <PolicyIdReference>, или <PolicySetIdReference>.

Для каждой роли должен быть определен один <PolicySet> Разрешения. В таком <PolicySet> должны содержаться элементы <Policy> и <Rule>, которые устанавливают типы доступов, разрешенные для Subjects,которым задана эта роль. <Target> элемента <PolicySet> и входящие в него или упомянутые в нем элементы <PolicySet>, <Policy>, и <Rule>, не должны ограничивать Subjects, к которым применим <PolicySet> Разрешения.

Если заданная роль наследует разрешения от одной или более младших ролей, то <PolicySet> Разрешения для заданной (старшей) роли должен включать в себя элемент <PolicySetIdReference> для каждой младшей роли. Каждый такой <PolicySetIdReference> должен ссылаться на <PolicySet> Разрешения, связанный с младшей ролью, от которой старшая роль наследует разрешения.

В <PolicySet> Разрешения может включать в себя HasPrivilegesOfRole <Policy>. У такого <Policy> должен быть элемент <Rule> с эффектом "Permit". Это правило должно разрешать любому субъекту выполнять действие с атрибутом, у которого AttributeId имеет значение &action;action-id, DataType имеет значение &xml;anyURI, и <AttributeValue> имеет значение &actions;hasPrivilegesOfRole, над ресурсом, имеющим атрибут, который является ролью, к которой применяется <PolicySet> Разрешения (например, AttributeId со значением &role;, DataType со значением &xml;anyURI, и <AttributeValue>, значением которого является URI конкретного ролевого атрибута). Заметим, что с ролевым атрибутом, который является атрибутом субъекта в <PolicySet> Роли <Target>, в HasPrivilegesOfRole <Policy>.обращаются, как с атрибутом ресурса.

Организация любого репозитария, используемого для стратегий и конфигураций PDP, должна гарантировать, что PDP никогда не сможет использовать <PolicySet> Разрешения в качестве исходной стратегии PDP.

8.6 Идентификаторы

В этом профиле определены следующие идентификаторы URN.

8.6.1 Идентификатор профиля

Следующий идентификатор должен использоваться в качестве идентификатора для этого профиля, если требуется идентификатор в форме URI.

```
urn:oasis:names:tc:xacml:2.0:profiles:rbac:core-hierarchical
```

8.6.2 Ролевой атрибут

Следующий идентификатор может использоваться в качестве `AttributeId` для ролевых атрибутов.

```
urn:oasis:names:tc:xacml:2.0:subject:role
```

8.6.3 SubjectCategory

Следующий идентификатор может использоваться в качестве `SubjectCategory` для атрибутов субъекта, идентифицирующих тот факт, что запрос исходит из органа задействования ролей.

```
urn:oasis:names:tc:xacml:2.0:subject-category:role-enablement-authority
```

8.6.4 Значения атрибута действия

Следующий идентификатор может использоваться в качестве атрибута `<AttributeValue>` со значением `&action;action-id` в `HasPrivilegesOfRole` `<Policy>`.

```
urn:oasis:names:tc:xacml:2.0:actions:hasPrivilegesOfRole
```

Следующий идентификатор может использоваться в качестве атрибута `<AttributeValue>` со значением `&action;action-id` в Role Assignment `<Policy>`.

```
urn:oasis:names:tc:xacml:2.0:actions:enableRole
```

9 Профиль множества ресурсов XACML

Оценка стратегии, выполняемая пунктом принятия решений XACML, или PDP, определяется в терминах отдельного запрашиваемого ресурса с решением об авторизации, содержащемся в отдельном элементе `<Result>` контекста запроса. В пункте осуществления стратегии, или PEP, однако, может понадобится представить контекст единого запроса о доступе к множеству ресурсов, и может понадобиться получение контекста единого запроса, в котором содержится единое решение об авторизации (элемент `<Result>`). Такой контекст запроса, например, может использоваться, для того чтобы избежать рассылки множества сообщений с запросами о принятии решения между PEP и PDP. С другой стороны, PEP может понадобиться представить контекст единого запроса ко всем узлам в иерархии, и может понадобиться получить единое решение об авторизации (элемент `<Result>`), который показывает, разрешен ли доступ ко всем запрашиваемым узлам. Такой контекст запроса может использоваться, например, если запросчик хочет получить доступ ко всему документу XML, ко всему поддереву элементов в таком документе, или ко всему системному справочнику файлов со всеми его подсправочниками и файлами.

В данной Рекомендации описывается три способа, с помощью которых PEP может запрашивать решения об авторизации для множества ресурсов в едином контексте запроса, и каким образом результат каждого такого решения об авторизации представляется в едином контексте ответа, который возвращается в PEP.

В данной Рекомендации также описывается два способа, с помощью которых PEP может сделать запрос о едином решении об авторизации в ответ на запрос для всех узлов в иерархии.

Поддержка для каждого из механизмов, описанных в этом профиле, является необязательной для соответствующих реализаций XACML.

Наиболее часто используемые атрибуты ресурсов сокращены следующим образом:

– атрибут "`resource-id`": атрибут ресурса со значением `AttributeId`:

  `"urn:oasis:names:tc:xacml:1.0:resource:resource-id"`

– атрибут "`scope`": атрибут ресурса со значением `AttributeId`:

  `"urn:oasis:names:tc:xacml:2.0:resource:scope"`

  Для получения более подробной информации об этом атрибуте смотрите пункт 9.3.

9.1 Запросы для множества ресурсов

Данный пункт является нормативным, но он необязательный.

Единый контекст запроса XACML может представлять запрос для доступа к множеству ресурсов, с желательным отдельным решением об авторизации для каждого ресурса. Синтаксис и семантика таких запросов и ответов устанавливается в этом пункте.

Элементы <Result>, полученные при поведении оценки запроса о доступе к множеству ресурсов, должны быть идентичны тем, которые были бы получены после оценки серии запросов, в каждом из которых запрашивался бы доступ только к одному из этих ресурсов. Каждый такой ресурс называется отдельный ресурс. Концептуальный контекст запроса, который соответствует каждому элементу <Result>, называется запросом об отдельном ресурсе. Значением ResourceId в элементе <Result> должен быть атрибут <AttributeValue> со значением "resource-id" в соответствующем запросе об отдельном ресурсе. Такое отображение первоначального контекста запроса, содержащее множество запросов о принятии решения об авторизации, в запросы об отдельных ресурсах и соответствующее отображение множества решений об авторизации во множество элементов <Result> в контексте единого запроса может выполняться обработчиком контекста, согласно данной Рекомендации. Данный профиль не требует, чтобы реализация оценки запроса о доступе к множеству ресурсов согласовывалась с предыдущей моделью или, чтобы создавались фактические запросы об отдельных ресурсах. В этом профиле только требуется, чтобы элементы <Result> были теми же самыми, как если бы использовалась предыдущая модель.

В следующих пунктах описаны три способа установления запросов о доступе к множеству ресурсов. В каждом способе установления запросов описаны запросы отдельных ресурсов, которые соотносятся с элементами <Result> в контексте запроса.

В едином контексте запроса XACML, представляемом пунктом PEP, может использоваться более одного из этих способов запроса доступа к множеству ресурсов в разных элементах <Resource>.

9.1.1 Узлы, идентифицируемые с помощью "scope"

Данный пункт является нормативным, но он необязательный.

В этом тексте описывается использование двух значений атрибута ресурса "scope" для установления запроса о доступе к множеству ресурсов в иерархии. Этот синтаксис может использоваться с любым иерархическим ресурсом, независимо от того, является ли он документом XML или нет.

9.1.1.1 Профиль URI

Следующие URI должны использоваться в качестве идентификаторов URI для той функциональности, которая установлена в этой части данного профиля. Первый идентификатор должен использоваться, если эта функциональность поддерживается для ресурсов XML, и второй идентификатор должен использоваться, если эта функциональность поддерживается для ресурсов, которые не являются документами XML:

```
urn:oasis:names:tc:xacml:2.0:profile:multiple:scope:xml
urn:oasis:names:tc:xacml:2.0:profile:multiple:scope:non-xml
```

9.1.1.2 Синтаксис первоначального контекста запроса

В элементе <Resource> первоначального контекста запроса XACML дожжен содержаться атрибут "scope" со значением либо "Children", либо "Descendants". Если запрашиваемые ресурсы находятся в документе XML, то должен присутствовать элемент <ResourceContent> и должен содержать полный документ XML, частью которого являются запрашиваемые элементы. Также, если запрашиваемые ресурсы находятся в документе XML, то выражение XPath, используемое в качестве значения атрибута "resource-id", должно оценивать комплект узлов, содержащих только один узел.

9.1.1.3 Семантика

Такой контекст запроса должен интерпретироваться как запрос о доступе для набора узлов в иерархии, соотнесенного с единым узлом, установленным в атрибуте "resource-id". Если значением атрибута "scope" является "Children", то каждый отдельный ресурс является одним узлом, указанным с помощью атрибута "resource-id" (или атрибутов, в которых единый ресурс имеет множество нормативных идентификаторов) и всеми своими непосредственными дочерними узлами. Если значением атрибута "scope" является "Descendants", то отдельный ресурс является одним узлом, указанным с помощью атрибута "resource-id" и всеми своими потомственными узлами.

Каждый отдельный запрос о ресурсе должен быть идентичен первоначальному контексту запроса с двумя исключениями: не должен присутствовать атрибут "scope" и элемент <Resource> должен представлять единый отдельный ресурс. В таком элементе <Resource> должен содержаться, по крайней мере, один атрибут "resource-id", и все значения для этих атрибутов должны быть уникальными нормативными идентичностями отдельного ресурса. Если в атрибуте "resource-id" в первоначальном контексте запроса содержится запрашивающая сторона, то в атрибутах "resource-id" отдельного запроса о ресурсе должна содержаться та же самая запрашивающая сторона. Если элемент <ResourceContent> присутствовал в первоначальном контексте запроса, то такой же элемент <ResourceContent> дожжен быть включен в каждый отдельный запрос о ресурсе.

Ни XACML, ни этот профиль не устанавливают каким образом обработчик контекста получает информацию, нужную для определения того, какие узлы являются дочерними, а какие потомственными в заданном узле, за исключением случая документа XML, в котором эта информация должна быть получена из элемента <ResourceContent>.

9.1.2 Узлы, идентифицируемые с помощью XPath

Данный пункт является нормативным, но он необязательный.

В этом пункте описывается использование выражения XPath в атрибуте "resource-id" вместе со значением "XPath-expression" в атрибуте "scope" для установления запроса о доступе к множеству узлов в документе XML. Этот синтаксис должен использоваться только с ресурсами, которые являются документами XML.

9.1.2.1 Профиль URI

Следующий URI должен использоваться в качестве идентификатора URI для функциональности, установленной в этой части данного профиля:

```
urn:oasis:names:tc:xacml:2.0:profile:multiple:xpath-expression
```

9.1.2.2 Первоначальный контекст запроса

В элементе <Resource> первоначального контекста запроса XACML должен содержаться элемент <ResourceContent> и атрибут "resource-id" с DataType со значением "urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression" таким, чтобы <AttributeValue> атрибута "resource-id" было выражение XPath, которое оценивается как набор узлов, представляющий множество узлов в элементе <ResourceContent>. В элементе <Resource> должен содержаться атрибут "scope" со значением "XPath-expression".

9.1.2.3 Семантика

Такой контекст запроса должен интерпретироваться как запрос о доступе для множества узлов, представляемых с помощью <AttributeValue> атрибута "resource-id". Каждый такой узел должен представлять отдельный ресурс.

Каждый отдельный запрос о ресурсе должен быть идентичным первоначальному контексту запроса с двумя исключениями: атрибут "scope" не должен присутствовать и значением атрибута "resource-id" должно быть выражение XPath, которое оценивается как единый узел в элементе <ResourceContent>. Такой узел должен быть отдельным ресурсом. Если в атрибуте "resource-id" первоначального контекста запроса содержится запрашивающая сторона, то в атрибуте "resource-id" в отдельном запросе о ресурсе должна содержаться та же запрашивающая сторона.

9.1.3 Множество элементов <Resource>

Данный пункт является нормативным, но он необязательный.

В этом пункте описывается использование множества элементов <Resource> в контексте запроса для установления запроса о доступе к множеству ресурсов. Этот синтаксис может использоваться с любым ресурсом или ресурсами, независимо от того, являются ли они документами XML или нет, и независимо от того, являются ли они иерархическими ресурсами или нет.

9.1.3.1 Профиль URI

Следующий URI должен использоваться в качестве идентификатора URI для функциональности, установленной в этой части данного профиля:

```
urn:oasis:names:tc:xacml:2.0:profile:multiple:multiple-resource-elements
```

9.1.3.2 Первоначальный контекст запроса

В контексте запроса XACML должно содержаться множество элементов <Resource>.

9.1.3.3 Семантика

Такой контекст запроса должен интерпретироваться как запрос о доступе ко всем ресурсам, установленным в отдельных элементах <Resource>. Каждый элемент <Resource> должен представлять один отдельный ресурс, если только этот элемент не использует другие механизмы, описанные в данном профиле.

Для каждого элемента <Resource> должен создаваться один отдельный запрос о ресурсе. Этот отдельный запрос о ресурсе должен быть идентичным первоначальному контексту запроса с одним исключением: должен присутствовать только один элемент <Resource>. Если в таком элементе <Resource> содержится атрибут "scope", имеющий любое другое значение, кроме "Immediate", то отдельный запрос о ресурсе должен в дальнейшем обрабатываться согласно соответствующей части данного профиля. В эту обработку входит разбиение одного отдельного запроса о ресурсе на другие отдельные запросы о ресурсе до проведения оценки пунктом PDP.

9.2 Запросы для всей иерархии

Данный пункт является нормативным, но он необязательный.

В некоторых случаях ресурс является иерархическим, но запрос о принятии решения об авторизации предназначается для запроса о доступе ко всем узлам внутри этого ресурса или ко всей подиерархии узлов внутри этого ресурса. Это может быть в случае, когда запрашивается доступ к документу XML с целью создания копии всего документа или когда запрашивается доступ ко всему системному справочнику файлов со всеми его подсправочниками и файлами. Желательно иметь единый <Result>, показывающий разрешено ли запросчику иметь доступ ко всему набору узлов.

Элемент <Result>, полученный с помощью оценки такого запроса о доступе, должен быть идентичным запросу, полученному с помощью следующего процесса. Выполняется оценка серии контекстов запроса, каждый из которых запрашивает доступ только к одному узлу в этой иерархии. <Decision> в едином <Result>, который возвращается в PEP, должен быть "Permit", если и только если все элементы <Result>, являющиеся результатом оценки отдельных узлов, содержащиеся в <Decision> имеют значение "Permit". Иначе, <Decision> в едином <Result>, возвращаемый в PEP должно быть "Deny". В данном профиле не требуется, чтобы реализация оценки запроса о доступе к такому иерархическому ресурсу была совместима с предыдущей моделью или, чтобы создавались фактические контексты запроса, соответствующие отдельным узлам в иерархии. В данном профиле только требуется, чтобы элемент <Result> был тем же самым, что и в случае использования предыдущей модели.

В следующих пунктах устанавливаются два синтаксиса для этой функциональности, один для использования с ресурсами, являющимися документами XML, и один для использования с ресурсами, не являющимися документами XML.

9.2.1 Ресурсы XML

Данный пункт является нормативным, но он необязательный.

В этом пункте описывается синтаксис для запроса о доступе ко всему документу XML или к элементу внутри этого документа со всеми его рекурсивными подэлементами.

9.2.1.1 Профиль URI

Следующий URI должен использоваться в качестве идентификатора для функциональности, установленной в этой части данного профиля:

```
urn:oasis:names:tc:xacml:2.0:profile:multiple:entire-hierarchy:xml
```

9.2.1.2 Первоначальный контекст запроса

В элементе <Resource> в первоначальном контексте запроса должен содержаться атрибут "scope" со значением "EntireHierarchy".

Элемент <Resource> в первоначальном контексте запроса должен содержать единый атрибут "resource-id" с DataType со значением "urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression", таким, чтобы <AttributeValue> оценивался, как набор узлов, представляющих только этот один узел в элементе <ResourceContent>.

В элементе <Resource> в первоначальном контексте запроса могут содержаться другие атрибуты.

9.2.1.3 Семантика

Элемент <Result> такого запроса должен быть эквивалентен элементу, который получается в ходе следующего процесса. Для каждого узла в запрашиваемой иерархии обработчик контекста должен создавать новый контекст запроса. В каждом таком контексте запроса должен содержаться единый элемент <Resource>, имеющий атрибут "resource-id" с DataType со значением "urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression" и значением, которое является выражением XPath, которое оценивается как набор узлов, содержащих только этот один узел в элементе <ResourceContent>. Обработчик контекста должен представить каждый такой новый контекст запроса в PDP для оценки и должен поддерживать тракт <Decision> в соответствующих элементах <Result>. Если и только если все новые контексты запросов оцениваются как "Permit", то единый <Result>, содержащий <Decision> со значением "Permit" должен быть помещен в контекст ответа, возвращаемого в PEP. Если какой-либо из новых контекстов запроса оценивается как "Deny", "Indeterminate" или "NotApplicable", то единый <Result>, содержащий <Decision> со значением "Deny", должен быть помещен в контекст ответа, возвращаемого в PEP.

9.2.2 Ресурсы, не являющиеся ресурсами XML

Данный пункт является нормативным, но он необязательный.

В этом пункте описывается синтаксис для запрашивания доступа ко всей иерархии узлов внутри иерархического ресурса, который не является документом XML.

9.2.2.1 Профиль URI

Следующий URI должен использоваться в качестве идентификатора для функциональности, установленной в этой части данного профиля:

```
urn:oasis:names:tc:xacml:2.0:profile:multiple:entire-hierarchy:non-xml
```

9.2.2.2 Первоначальный контекст запроса

В элементе <Resource> в первоначальном контексте запроса должен содержаться атрибут "scope" со значением "EntireHierarchy".

Элемент <Resource> в первоначальном контексте запроса должен содержать единый атрибут "resource-id", который представляет единый узел в иерархическом ресурсе.

В элементе <Resource> в первоначальном контексте запроса могут содержаться другие атрибуты.

Представление узлов в иерархическом ресурсе, установленное в этом профиле XACML для иерархических ресурсов в данной Рекомендации, может использоваться для представления идентичности единого узла.

9.2.2.3 Семантика

Элемент <Result> такого запроса должен быть эквивалентен элементу, который получается в ходе следующего процесса. Для каждого узла в запрашиваемой иерархии обработчик контекста должен создавать новый контекст запроса. В каждом таком контексте запроса должен содержаться единый элемент <Resource>, имеющий атрибут "resource-id" со значением, которое является идентичностью этого одного узла в иерархии. Обработчик контекста должен представить каждый такой новый контекст запроса в PDP для оценки и должен поддерживать тракт <Decision> в соответствующих элементах <Result>. Если и только если все новые контексты запросов оцениваются как "Permit", то единый <Result>, содержащий <Decision> со значением "Permit" должен быть помещен в контекст ответа, возвращаемого в PEP. Если какой-либо из новых контекстов запроса оценивается как "Deny", "Indeterminate" или "NotApplicable", то единый <Result>, содержащий <Decision> со значением "Deny", должен быть помещен в контекст ответа, возвращаемого в PEP.

Ни XACML, ни этот профиль не устанавливают каким образом обработчик контекста получает информацию, нужную для определения того, какие узлы являются потомками первоначально заданного узла или как представлять идентичность каждого такого узла. Представление узлов в иерархическом ресурсе, установленное в этом профиле XACML для иерархических ресурсов в данной Рекомендации, может использоваться для представления идентичности каждого такого узла.

9.3 Новые идентификаторы атрибутов

Следующий идентификатор используется, как AttributeId атрибута ресурса, который показывает область применения (атрибут "scope") запроса о доступе в едином элементе <Resource> контекста запроса.

```
urn:oasis:names:tc:xacml:2.0:resource:scope
```

У этого атрибута должен быть DataType со значением "http://www.w3.org/2001/XMLSchema#string".

Действительные значения для этого атрибута перечислены ниже и в пункте 7.5. Реализация может поддерживать любой подмножество этих значений, включая пустой набор.

- "Immediate" – элемент <Resource> относится к неиерархическому ресурсу или к единому узлу в иерархическом ресурсе. Это значение по умолчанию, если нет ни одного атрибута "scope". Элемент <Resource> должен обрабатываться в соответствии с пунктом 7.
- "Children" – элемент <Resource> относится к множеству ресурсов в иерархии. Этот набор ресурсов состоит из единого узла, описываемого атрибутом ресурса "resource-id" и из всех непосредственных дочерних узлов в этой иерархии. Элемент <Resource> должен обрабатываться в соответствии с пунктом 9.1.1 этого профиля.
- "Descendants" – элемент <Resource> относится к множеству ресурсов в иерархии. Этот набор ресурсов состоит из единого узла, описываемого атрибутом ресурса "resource-id" и из всех потомков узлов в этой иерархии. Элемент <Resource> должен обрабатываться в соответствии с пунктом 9.1.1 этого профиля.
- "XPath-expression" – элемент <Resource> относится к множеству ресурсов. Этот набор ресурсов состоит из узлов в наборе узлов, описываемых атрибутом ресурса "resource-id". Каждый такой узел должен содержаться в элементе <ResourceContent> элемента <Resource>. Элемент <Resource> должен обрабатываться в соответствии с пунктом 9.1.2 этого профиля.

– "EntireHierarchy" – элемент <Resource> относится к единому ресурсу. Этот ресурс состоит из узла, описываемого атрибутом ресурса "resource-id" вместе со всеми потомками узла. Все эти узлы должны быть узлами в документе XML, который содержится в элементе <ResourceContent> элемента <Resource>. Элемент <Resource> должен обрабатываться в соответствии с пунктом 9.2. этого профиля.

9.4 Новые идентификаторы профиля

Следующие значения URI должны использоваться в качестве идентификаторов URI для функциональности, установленной в разных пунктах этого профиля:

– атрибут "scope" со значением "children" или "descendants" в <Resource>: ресурсы XML

```
urn:oasis:names:tc:xacml:2.0:profile:multiple:scope:xml
```

– атрибут "scope" со значением "children" или "descendants" в <Resource>: ресурсы, не являющиеся ресурсами XML

```
urn:oasis:names:tc:xacml:2.0:profile:multiple:scope:non-xml
```

– выражение XPath в атрибуте "resource-id"

```
urn:oasis:names:tc:xacml:2.0:profile:multiple:xpath-expression
```

– множество элементов <Resource>

```
urn:oasis:names:tc:xacml:2.0:profile:multiple:multiple-resource-
elements
```

– запросы для всей иерархии: ресурсы XML

```
urn:oasis:names:tc:xacml:2.0:profile:multiple:entire-hierarchy:xml
```

– запросы для всей иерархии: ресурсы, не являющиеся ресурсами XML

```
urn:oasis:names:tc:xacml:2.0:profile:multiple:entire-hierarchy:non-
xml
```

10 Профиль SAML 2.0 языка XACML

В этом пункте определяется профиль относительно использования SAML 2.0 (смотрите Рекомендацию МСЭ-Т X.1141) для защиты, транспортировки и запроса экземпляров схемы XACML и другой информации, которая требуется для реализации XACML.

SAML является структурой на основе XML для обмена информацией, касающейся безопасности. Эта информация, касающаяся безопасности, выражена в форме утверждений о субъектах, где субъект является объектом (либо человек, либо компьютер), имеющим идентичность в каком-либо домене безопасности. Единое утверждение может содержать несколько разных внутренних заявлений об аутентификации, авторизации и атрибутах. В SAML определяется протокол, с помощью которого клиенты могут запросить утверждения у органов SAML и получить от них ответ. Этот протокол, состоящий из форматов сообщений запросов и ответов на основе XML, может быть связан с большим количеством разных базовых протоколов связи и транспортировки; в SAML в настоящее время определен один протокол связывания SOAP посредством HTTP. При создании ответов органы SAML могут использовать различные источники информации, такие как внешние хранилища стратегий и утверждений, которые были получены в качестве входных данных в запросах. В SAML определяются элементы утверждения, субъекты, условия, советы и заявления.

В этом пункте используются шесть типов запросов и заявлений:

1) AttributeQuery: стандартный запрос SAML, использующийся для запрашивания одного или более атрибутов у органа атрибутов.
2) AttributeStatement: стандартное заявление SAML, в котором содержится один или более атрибутов. Это заявление может использоваться в ответе SAML от органа атрибутов или оно может использоваться в утверждении SAML в качестве формата для хранения атрибутов в репозитории атрибутов.
3) XACMLPolicyQuery: расширение запроса SAML, определенное в этом профиле. Оно используется для запрашивания одной или более стратегий у пункта управления стратегией.
4) XACMLPolicyStatement: расширение заявления SAML, определенное в этом профиле. Оно может использоваться в ответе SAML от пункта управления стратегией или оно может использоваться в утверждении SAML в качестве формата для хранения стратегий в репозитарии стратегий.
5) XACMLAuthzDecisionQuery: расширение запроса SAML, определенное в этом профиле. Оно используется пунктом PEP для запроса решения об авторизации у PDP XACML.

6) `XACMLAuthzDecisionStatement`: расширение заявления SAML, определенное в этом профиле. Оно может использоваться в ответе SAML от PDP XACML. Оно также может использоваться в утверждении SAML, которое используется как рекомендательное, но не является частью, определяемой в настоящее время модели использования XACML.

Следующая диаграмма (рисунок 10-1) иллюстрирует модель использования XACML и сообщения, использующиеся для связи между различными компонентами. Не все компоненты будут использоваться в каждой реализации.

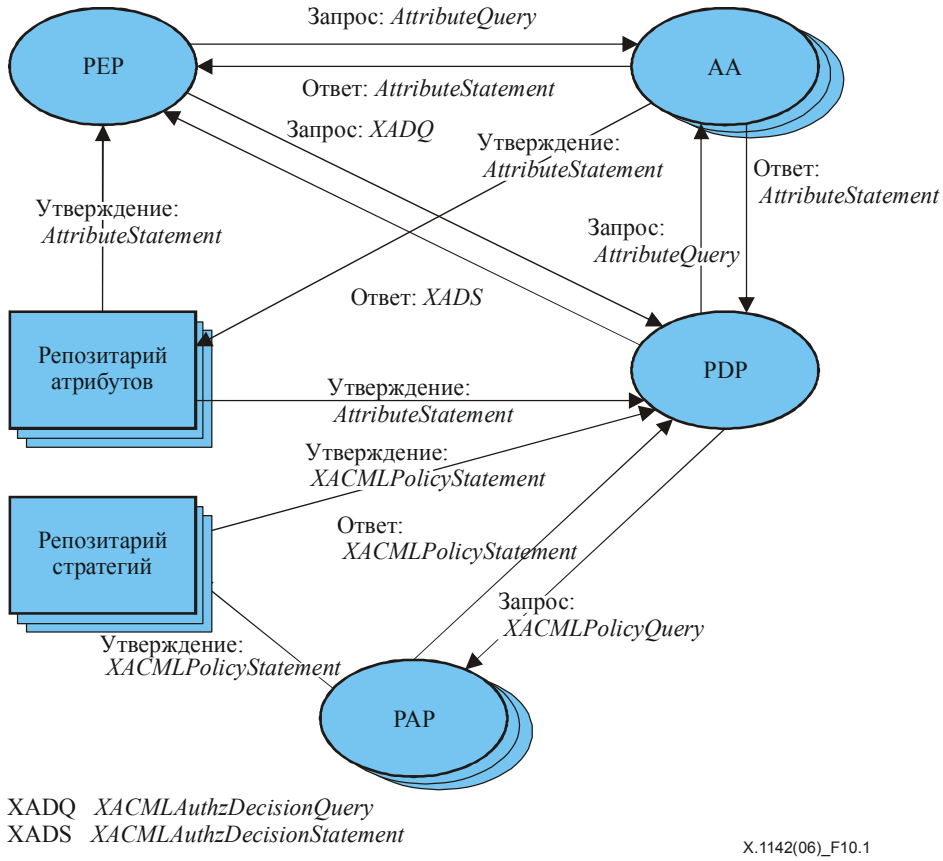

Рисунок 10-1/X.1142 – Модель использования XACML

В этом пункте описываются все элементы схемы утверждений и запросов и описывается как их использовать. В нем также описываются некоторые другие аспекты использования SAML с XACML. В данной Рекомендации не требуются изменения или расширения к XACML, но определяются расширения к SAML.

С целью улучшения удобочитаемости для примеров в этом профиле принято использование следующих описаний внутренних объектов XML:

```
^lt;!ENTITY saml "urn:oasis:names:tc:SAML:2.0:assertion"
^lt;!ENTITY samlp "urn:oasis:names:tc:SAML:2.0:protocol"
^lt;!ENTITY xacml "urn:oasis:names:tc:xacml:2.0:"
^lt;!ENTITY xacml-context "urn:oasis:names:tc:xacml:2.0:context:schema:os"
^lt;!ENTITY xml "http://www.w3.org/2001/XMLSchema#"
^lt;!ENTITY subject-id "urn:oasis:names:tc:xacml:1.0:subject:subject-id"
^lt;!ENTITY resource "urn:oasis:names:tc:xacml:1.0:resource:"
^lt;!ENTITY resource-id "urn:oasis:names:tc:xacml:1.0:resource:resource-id"
^lt;!ENTITY action-id "urn:oasis:names:tc:xacml:1.0:action:action-id"
^lt;!ENTITY environment "urn:oasis:names:tc:xacml:1.0:environment:"
^lt;!ENTITY current-dateTime
    "urn:oasis:names:tc:xacml:1.0:environment:current-dateTime"
```

Например, "`&xml;#string`" эквивалентно http://www.w3.org/2001/XMLSchema#string. Пространство имен, связанное со схемой XACML, которое расширяет схему утверждений SAML:

```
xacml-saml="urn:oasis:names:tc:xacml:2.0:saml:assertion:schema:os"
```

Пространство имен, связанное со схемой XACML, которое расширяет схему протоколов SAML:

```
xacml-samlp="urn:oasis:names:tc:xacml:2.0:saml:protocol:schema:os"
```

10.1 Отображение атрибутов SAML и XACML

Схема утверждений SAML определяет утверждение атрибута. Схема протокола SAML определяет `AttributeQuery`, используемый для запрашивания объектов утверждений атрибутов, и ответ, в котором содержатся запрашиваемые экземпляры. Системы, использующие XACML, могут использовать экземпляры этих элементов SAML, передающих и сохраняющих атрибуты SAML. Системы, использующие XACML, могут использовать протокол `AttributeQuery` SAML, для того чтобы запросить экземпляры атрибутов SAML. Для использования в контексте запроса XACML, атрибут SAML должен быть отображен в атрибут XACML.

Утверждением атрибута SAML является экземпляр `<saml:Assertion>`, в котором содержится один или более экземпляров `<saml:AttributeStatement>`, в каждом из которых может содержаться один или более экземпляров `<saml:Attribute>`.

Для использования в контексте запроса XACML, каждый атрибут SAML в утверждении атрибута SAML должен согласовываться с *XACML Attribute profile*, пространством имен `urn:oasis:names:tc:SAML:2.0:profiles:attribute:XACML`, в Рекомендации МСЭ-Т X.1141.

`<xacml-context:Attribute>` должен быть образован от соответствующего элемента `<saml:Attribute>` в утверждении атрибута SAML следующим образом.

- Атрибут XML `AttributeId` XACML
  - Должно использоваться полностью подходящее значение атрибута XML `<saml:Attribute>` `Name`.
- Атрибут XML `DataType` XACML
  - Должно использоваться полностью подходящее значение атрибута XML `<saml:Attribute>` `DataType`. Если атрибут XML `<saml:Attribute>` `DataType` пропущен, то атрибутом XML `DataType` XACML должно быть http://www.w3.org/2001/XMLSchema#string.
- Атрибут XML `Issuer` XACML
  - Должно использоваться строковое значение элемента `<saml:Issuer>` из утверждения атрибута SAML.
- `<xacml-context:AttributeValue>`
  - Значение `<saml:AttributeValue>` должно использоваться, в качестве значения элемента `<xacml-context:AttributeValue>`.

Каждый экземпляр `<saml:Attribute>` отображается в единый элемент `<xacml-context:Attribute>`. Не все экземпляры `<saml:Attribute>` в утверждении атрибута SAML нужно отображать; экземпляры атрибута SAML, которые нужно отобразить, могут выбираться с помощью механизмов, которые здесь не устанавливаются. `Issuer` элемента `<saml:Assertion>` используется в качестве `Issuer` для каждого создаваемого элемента `<xacml-context:Attribute>`.

`<xacml-context:Attribute>` созданный из `<saml:Assertion>`, должен помещаться в элемент `<xacml-context:Resource>`, `<xacml-context:Subject>`, `<xacml-context:Action>`, или `<xacml-context:Environment>`, относящийся к объекту, которым является `<saml:Subject>` в утверждении атрибута SAML. Например, если в субъекте утверждения атрибута SAML содержится элемент `<saml:NameIdentifier>`, и значение `NameIdentifier` сопоставимо со значением `<xacml-context:Attribute>`, у которого имеется `AttributeId` со значением `&resource;resource-id`, то экземпляры `<xacml-context:Attribute>`, созданные из экземпляров `<saml:Attribute>` в этом утверждении атрибута SAML должны помещаться в элемент `<xacml-context:Resource>`. Если `<xacml-context:Attribute>` помещается в элемент `<xacml-context:Subject>`, то атрибут XML `SubjectCategory` XACML должен также согласовываться с объектом, который является субъектом `<saml:Assertion>`.

Объект, выполняющий отображение, должен гарантировать соблюдение семантики, определенной с помощью SAML для элементов в `<saml:Assertion>`. Объекту отображения не нужно самому выполнять проверку этой семантики, но он должен гарантировать, что такая проверка выполняется до того, как какой-либо `<xacml:Attribute>`, созданный из `<saml:Assertion>`, используется пунктом PDP XACML. В проверку этой семантики входит, но этим не ограничивается, следующее:

- Любые атрибуты XML `NotBefore` и `NotOnOrAfter` в `<saml:Assertion>` должны быть действительны с учетом `<xacml:Request>`, в котором используется, полученный из SAML `<xacml:Attribute>`. Это означает, что значения атрибутов XML `NotBefore` и `NotOnOrAfter` должны согласовываться со значениями `&environment;current-time`, `&environment;current-date`, и `&environment:current-dateTime` `<xacml:Attribute>`, связанными с `<xacml:Request>`.

- Объект, выполняющий отображение, должен гарантировать соблюдение семантики, определенной SAML для любых элементов <saml:AudienceRestrictionCondition> или <saml:DoNotCacheCondition>.
- Если элемент <ds:Signature> появляется в <saml:Assertion>, то объект, выполняющий отображение, должен гарантировать действительность подписи и согласованность элемента <Issuer> SAML с любым значением <ds:X509IssuerName> в этой подписи. Должны соблюдаться руководящие указания относительно цифровых подписей в Рекомендации МСЭ-Т X.1141.

10.2 Решения об авторизации

В SAML 2.0 определяется рудиментарный AuthzDecisionQuery (смотрите Рекомендацию МСЭ-Т X.1141). AuthzDecisionQuery SAML не в состоянии передавать всю ту информацию, которую PDP XACML может принимать, как часть его контекста запроса. Точно так же, AuthzDecisionStatement SAML не способен передавать всю ту информацию, которая содержится в контекст е запроса XACML.

Для того чтобы позволить PEP использовать синтаксис запроса и ответа SAML с полной поддержкой для синтаксиса контекста запроса и контекста ответа XACML, в данной Рекомендации определяется два расширения SAML:

- <xacml-samlp:XACMLAuthzDecisionQuery> является запросом SAML, который расширяет схему протокола SAML (смотрите Рекомендацию МСЭ-Т X.1141). Это позволяет PEP представлять контекст запроса XACML в запросе SAML, вместе с другой информацией.
- <xacml-saml:XACMLAuthzDecisionStatement> является утверждением SAML, которое расширяет схему утверждений SAML (смотрите Рекомендацию МСЭ-Т X.1141). Это позволяет PDP XACML возвращать контекст ответа XACML в ответе в <XACMLAuthzDecisionStatement>, вместе с другой информацией. Это также позволяет сохранять или передавать контекст ответа XACML в форме утверждения SAML.

10.2.1 Элемент <XACMLAuthzDecisionQuery>

Элемент <XACMLAuthzDecisionQuery> может использоваться PEP для запроса решения об авторизации у PDP XACML. Это позволяет запросу SAML передавать экземпляр контекста запроса XACML.

```
<xs:element name="XACMLAuthzDecisionQuery" type="XACMLAuthzDecisionQueryType"/>
<xs:complexType name="XACMLAuthzDecisionQueryType">
  <xs:complexContent>
   <xs:extension base="samlp:RequestAbstractType">
    <xs:sequence>
        <xs:element ref="xacml-context:Request"/>
    </xs:sequence>
    <xs:attribute name="InputContextOnly" type="boolean" use="optional" default="false"/>
    <xs:attribute name="ReturnContext" type="boolean" use="optional" default="false"/>
   </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Элемент <XACMLAuthzDecisionQuery> является элементом составного типа **XACMLAuthzDecisionQueryType**. Этот элемент является альтернативой <samlp:AuthzDecisionQuery>, определенному в SAML, что позволяет PEP использовать полные возможности PDP XACML.

В элементе <XACMLAuthzDecisionQuery> содержатся следующие атрибуты и элементы XML:

– InputContextOnly ["false" по умолчанию]

  Этот атрибут XML управляет источниками информации, которыми разрешено пользоваться PDP для создания решения об авторизации. Если этот атрибут XML имеет значение "true", то решение об авторизации должно быть принято исключительно на базе той информации, которая содержится в <XACMLAuthzDecisionQuery>; не нужно использовать внешние атрибуты. Если этот атрибут XML имеет значение "false", то решение об авторизации может быть принято на базе внешних атрибутов, не содержащихся в <XACMLAuthzDecisionQuery>.

– ReturnContext ["false" по умолчанию]

  Этот атрибут XML позволяет PEP сделать запрос о том, чтобы элемент <xacml-context:Request> был включен в <XACMLAuthzDecisionStatement>, образующийся в результате запроса. Он также управляет содержанием этого элемента <xacml-context:Request>.

Если этот атрибут XML имеет значение "true", то PDP должен включить элемент <xacml-context:Request> в элемент <XACMLAuthzDecisionStatement> в <XACMLResponse>. В этот элемент <xacml-context:Request> должны быть включены все те атрибуты, которые поставляются с помощью PEP в <XACMLAuthzDecisionQuery>, и которые использовались в создании решения об авторизации. PDP может включать дополнительные атрибуты в этот элемент <xacml-context:Request>, такие как внешние атрибуты, полученные PDP и использующиеся в создании решения об авторизации или другие известные PDP атрибуты, которые могут быть полезными для PEP в создании последующих запросов <XACMLAuthzDecisionQuery>.

Если этот атрибут XML имеет значение "false", то PDP не должен включать элемент <xacml-context:Request> в элемент <XACMLAuthzDecisionStatement> в <XACMLResponse>.

– <xacml-context:Request> [Обязательный]

Контекст запроса XACML.

10.2.2 Элемент <XACMLAuthzDecisionStatement>

<XACMLAuthzDecisionStatement> может использоваться PDP XACML для возвращения ответа SAML, содержащего контекст ответа XACML, в PEP в ответ на <XACMLAuthzDecisionQuery>. Он также может использоваться в утверждении SAML в качестве формата для хранения решений об авторизации в репозитарии.

```
<xs:element name="XACMLAuthzDecisionStatement" type="xacml-
saml:XACMLAuthzDecisionStatementType"/>
<xs:complexType name="XACMLAuthzDecisionStatementType">
  <xs:complexContent>
     <xs:extension base="saml:StatementAbstractType">
        <xs:sequence>
           <xs:element ref="xacml-context:Response"/>
              <xs:element ref="xacml-context:Request" MinOccurs="0"/>
        </xs:sequence>
     </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Элемент <XACMLAuthzDecisionStatement> является элементом составного типа **XACMLAuthzDecisionStatementType**. Этот элемент является альтернативой <samlp:AuthzDecisionStatement>, определенному в SAML, что позволяет утверждению SAML иметь в своем составе полное содержание ответов от PDP XACML.

В элементе <XACMLAuthzDecisionStatement> содержатся следующие элементы:

– <xacml-context:Response> [Обязательный]

Контекст ответа XACML, созданный PDP XACML в ответ на <XACMLAuthzDecisionQuery>.

– <xacml-context:Request> [Необязательный]

<xacml-context:Request>, содержащий атрибуты XACML, возвращенные PDP XACML в ответ на <XACMLAuthzDecisionQuery>. Этот элемент должен быть включен, если атрибут ReturnResponse XML в <XACMLAuthzDecisionQuery> имеет значение "true". Этот элемент не должен быть включен, если атрибут ReturnResponse XML в <XACMLAuthzDecisionQuery> имеет значение "false".

10.3 Стратегии

В XACML определяются два элемента схемы стратегии: <Policy> и <PolicySet>. В SAML не должны определяться какие-либо протоколы или схемы утверждений для стратегий. В этом пункте определяются новые расширения SAML для элементов <XACMLPolicyQuery> и <XACMLPolicyStatement>. Экземпляры этих новых элементов могут использоваться для запроса, передачи и хранения экземпляров <Policy> и <PolicySet> XACML.

10.3.1 Элемент <XACMLPolicyQuery>

Элемент <XACMLPolicyQuery> используется PDP для запроса одной или более стратегий XACML или экземпляров PolicySet у онлайнового пункта управления стратегией как части запроса SAML.

```
<xs:element name="XACMLPolicyQuery" type="XACMLPolicyQueryType"/>
<xs:complexType name="XACMLPolicyQueryType">
   <complexContent>
         <xs:extension base="samlp:RequestAbstractType">
```

```
                <xs:choice minOccurs="0" maxOccurs="unbounded">
                      <xs:element ref="xacml-context:Request"/>
                      <xs:element ref="xacml:Target"/>
                      <xs:element ref="xacml:PolicySetIdReference"/>
                      <xs:element ref="xacml:PolicyIdReference"/>
                </xs:choice>
          </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

Элемент <XACMLPolicyQuery> является элементом составного типа **XACMLPolicyQueryType**.

В элементе <XACMLPolicyQuery> содержится один или более следующих элементов:

– <xacml-context:Request> [Любое количество]

  Доставляет контекст запроса XACML. Все стратегии XACML и экземпляры PolicySet, применимые к этому запросу, должны быть возвращены.

– <xacml:Target> [Любое количество]

  Доставляет элемент <Target> XACML. Все стратегии XACML и экземпляры PolicySet, применимые к этому <Target>, должны быть возвращены.

– <xacml:PolicySetIdReference> [Любое количество]

  Идентифицирует <PolicySet> XACML, который должен быть возвращен.

– <xacml:PolicyIdReference> [Любое количество]

  Идентифицирует <Policy> XACML, который должен быть возвращен.

10.3.2 Элемент <XACMLPolicyStatement>

<XACMLPolicyStatement> используется пунктом управления стратегией для возвращения одного или более экземпляров <Policy> или <PolicySet> XACML в ответе SAML в <XACMLPolicyQuery> запроса SAML. Элемент <XACMLPolicyStatement> может также использоваться в утверждении SAML в качестве формата для хранения <XACMLPolicyStatement> в репозитарии.

```
<xs:element name="XACMLPolicyStatement" type="xacml-
saml:XACMLPolicyStatementType"/>
<xs:complexType name="XACMLPolicyStatementType">
   <xs:complexContent>
          <xs:extension base="saml:StatementAbstractType">
                <xs:choice minOccurs="0" maxOccurs=unbounded">
                      <xs:element ref="xacml:Policy"/>
                      <xs:element ref="xacmlPolicySet"/>
                </xs:choice>
                </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

Элемент <XACMLPolicyStatement> является элементом составного типа **XACMLPolicyStatementType**.

В элементе <XACMLPolicyStatement> содержатся следующие элементы. Если <XACMLPolicyStatement> выпущен в ответ на <XACMLPolicyQuery> и отсутствуют экземпляры <xacml:Policy> или <xacml:PolicySet>, отвечающие требованиям технических условий связанного <XACMLPolicyQuery>, то не должно быть элементов в <XACMLPolicyStatement>.

– <xacml:Policy> [Любое количество]

  Экземпляр <xacml:Policy>, отвечающий техническим требованиям связанного <XACMLPolicyQuery>, если таковой имеется.

– <xacml:PolicySet> [Любое количество]

  Экземпляр <xacml:PolicySet>, отвечающий техническим требованиям связанного <XACMLPolicyQuery>, если таковой имеется.

10.4 Элемент <saml:Assertion>

<XACMLAuthzDecisionStatement>, <XACMLPolicyStatement> или <saml:AttributeStatement> стандарта SAML должен быть инкапсулирован в <saml:Assertion>, который может быть снабжен подписью.

Большинство компонентов элемента <saml:Assertion> полностью установлены в Рекомендации МСЭ-Т X.1141. Следующие элементы и атрибуты XML устанавливаются здесь более полно для использования с типами утверждений SAML, определенными и использованными в этом профиле.

За исключением установленных здесь, этим профилем не вводится никаких требований или ограничений на информацию в элементе <saml:Assertion>.

10.4.1 Элемент <saml:Issuer>

Элемент <saml:Issuer> является элементом, нужным для хранения информации об "Органе SAML, который предъявляет требование(я) в утверждении".

Для поддержания цифровых подписей третьих лиц в этом профиле не требуется, чтобы идентичность, предоставленная в элементе <saml:Issuer>, согласовывалась бы с идентичностью подписавшей стороны. От стороны-доверителя зависят соответствующие доверительные отношения с органом, который подписывает <saml:Assertion>.

Если <saml:AttributeAssertion> используется для создания атрибута XACML, то строковое значение элемента <saml:Issuer> будет использоваться, как значение атрибута XACML Issuer XML, поэтому значение SAML должно устанавливаться с учетом сказанного.

10.4.2 Элемент <ds:Signature>

Элемент <ds:Signature> является необязательным элементом для хранения "Подписи XML, которая аутентифицирует утверждение".

Элемент <ds:Signature> может использоваться в утверждении, используемом с утверждением XACML. Для поддержания цифровых подписей третьих лиц в этом профиле не требуется, чтобы идентичность, предоставленная в элементе <saml:Issuer>, согласовывалась бы с идентичностью подписавшей стороны. От стороны-доверителя зависят соответствующие доверительные отношения с органом, который подписывает <saml:Assertion>.

Сторона-доверитель должна подтверждать любую подпись, включенную в утверждение, и не должна использовать информацию, полученную из утверждения до тех пор пока подпись не будет успешно подтверждена.

10.4.3 Элемент <saml:Subject>

Элемент <saml:Subject> является необязательным элементом для хранения "Субъекта заявления(ий) в утверждении".

Элемент <saml:Subject> должен быть включен в утверждение, в котором содержится <XACMLAuthzDecision> или <XACMLPolicy>.

В <saml:AttributeAssertion>, который должен отображаться в атрибут XACML, элемент <saml:Subject> должен содержать идентичность объекта, с которым связаны этот атрибут и его значение. Для атрибута XACML <Subject> эта идентичность должна согласовываться со значением любого атрибута &subject-id; XACML, который появляется в том же самом элементе <Subject>.

Для атрибута <Resource> XACML эта идентичность должна согласовываться со значением любого атрибута &resource-id; XACML, который появляется в том же самом элементе <Resource>. Для атрибута <Action> XACML эта идентичность должна согласовываться со значением атрибута &action-id; XACML, который появляется в том же самом элементе <Action>. Для атрибута <Environment> XACML эта идентичность должна согласовываться со значением любого атрибута XACML, который появляется в том же самом элементе <Environment> и предоставляет идентичность среды.

10.4.4 Элемент <saml:Conditions>

Элемент <saml:Conditions> является необязательным элементом, используемым для "условий, которые должны учитываться при оценке действительности и/или использовании утверждения".

В элементе <saml:Conditions> должны содержаться атрибуты NotBefore и NotOnOrAfter XML для установления ограничений на действительность утверждения. Если эти атрибуты XML присутствуют, то сторона-доверитель должна гарантировать, что информация, полученная из утверждения, используется PDP для оценки стратегий, только если значение атрибута источника &current-dateTime; контекста запроса содержится внутри периода действительности, определенного в утверждении.

10.5 Элемент <samlp:RequestAbstractType>

<XACMLAuthzDecisionQuery> или <XACMLPolicyQuery> должен быть инкапсулирован в элементе <samlp:RequestAbstractType>, который может быть снабжен подписью.

Большинство компонентов <samlp:RequestAbstractType> полностью определены в Рекомендации МСЭ-Т X.1141. Для использования с типами запросов SAML, определенных и использованных в этом профиле, элемент <saml:Issuer> и элемент <ds:Signature> должны использоваться таким же образом, как это определено в предыдущем пункте. За исключением установленных здесь, этим профилем не вводится никаких требований или ограничений на информацию в элементе <samlp:RequestAbstractType>.

10.5.1 Элемент <saml:Issuer>

Смотрите пункт 10.4.1, элемент <saml:Issuer>.

10.5.2 Элемент <ds:Signature>

Смотрите пункт 10.4.2, элемент <ds:Signature>.

10.6 Элемент <samlp:Response>

<XACMLAuthzDecisionStatement> или <XACMLPolicyStatement> должен быть инкапсулирован в элементе <samlp:Response> который может быть снабжен подписью.

Большинство компонентов <samlp:Response> полностью определены в Рекомендации МСЭ-Т X.1141. Следующие элементы и атрибуты XML устанавливаются здесь более полно для использования с типами утверждений SAML, определенными и использованными в этом профиле. За исключением установленных здесь, этим профилем не вводится никаких требований или ограничений на информацию в элементе <samlp:Response>.

10.6.1 Элемент <samlp:Issuer>

Смотрите пункт 10.4.1, элемент <saml:Issuer>.

10.6.2 Элемент <ds:Signature>

Смотрите пункт 10.4.2, элемент <ds:Signature>.

10.6.3 Элемент <samlp:StatusCode>

Элемент <samlp:StatusCode> является компонентом элемента <samlp:Status> в <samlp:Response>.

10.6.3.1 Ответ на <XACMLAuthzDecisionQuery>

В ответе на запрос <XACMLAuthzDecisionQuery>, атрибут XML Value <samlp:StatusCode> должен зависеть от элемента <xacml:StatusCode> из элемента <xacml:Status> решения об авторизации следующим образом:

1) urn:oasis:names:tc:SAML:2.0:status:Success

   Это значение для атрибута XML Value <samlp:StatusCode> должно использоваться, если и только если значением <xacml:StatusCode> является urn:oasis:names:tc:xacml:1.0:status:ok.

2) urn:oasis:names:tc:SAML:2.0:status:Requester

   Это значение для атрибута XML Value <samlp:StatusCode> должно использоваться, если значением <xacml:StatusCode> является urn:oasis:names:tc:xacml:1.0:status:missing-attribute или, если значением <xacml:StatusCode> является urn:oasis:names:tc:xacml:1.0:status:syntax-error из-за ошибки синтаксиса в <xacml:Request>.

3) urn:oasis:names:tc:SAML:2.0:status:Responder

   Это значение для атрибута XML Value <samlp:StatusCode> должно использоваться, если значением <xacml:StatusCode> является urn:oasis:names:tc:xacml:1.0:status:syntax-error из-за ошибки синтаксиса в <xacml:Policy> или <xacml:PolicySet>. Заметим, что не все ошибки синтаксиса в стратиях будут обнаружены в процессе обработки конкретного запроса, поэтому не обо всех ошибках синтаксиса стратегии будет сообщаться таким образом.

4) urn:oasis:names:tc:SAML:2.0:status:VersionMismatch

   Это значение для атрибута XML Value <samlp:StatusCode> должно использоваться только, если интерфейс SAML в PDP не поддерживает версию сообщения запроса SAML, использованного в этом запросе.

10.6.3.2 Ответ на < XACMLPolicyQuery >

В ответе на запрос `<XACMLPolicyQuery>` атрибут XML Значение `<samlp:StatusCode>` должен быть таким, как установлено в Рекомендации МСЭ-Т X.1141.

11 Профиль цифровой подписи XML

В этом пункте предоставляется профиль для использования с W3C Signature:2002 для предоставления аутентификации и защиты целостности для экземпляров схемы XACML.

Цифровая подпись полезна для аутентификации и защиты целостности только, если в подписанную информацию включена спецификация идентичности подписанта и спецификация периода, в течение которого объект подписанных данных должен быть признан действительным. Сам язык XACML не определяет формат для такой информации, так как XACML собирается использовать другие стандарты для функций, кроме фактической спецификации и оценки стратегий, запросов и ответов контроля доступа.

Один подходящий формат определен в SAML. Профиль для использования SAML с экземплярами схемы XACML определен в пункте 10. Таким образом в этом профиле рекомендуется использовать экземпляры схемы XACML в Утверждениях, Запросах и Ответах SAML, которые затем могут быть подписаны цифровой подписью, как установлено в Рекомендации МСЭ-Т X.1141.

11.1 Использование SAML

В этом профиле рекомендуется использовать экземпляры схемы XACML, встроенные в Утверждения, Запросы и Ответы SAML, как описано в пункте 10. Такие объекты SAML должны быть подписаны цифровой подписью, как описано в 8.4/X.1141, *SAML and XML Signature Syntax and Processing*.

11.2 Канонизация

Для того чтобы цифровая подпись могла быть подтверждена стороной-доверителем, подписанный поток байтов должен быть идентичен подтверждаемому потоку байтов. Для гарантий этого требования, подписываемый документ XML должен быть канонизирован (смотрите W3C Canonicalization:2002). В Рекомендации МСЭ-Т X.1141 устанавливается использование исключающей канонизации (смотрите W3C Canonicalization:2002).

11.2.1 Элементы пространства имен в объектах данных XACML

Любой объект данных XACML, который нужно подписать, должен установить все элементы пространства имен, используемых в этом объекте данных. Если это не сделано, то объект данных будет привлекать определения пространства имен от предшествующих элементов этого объекта данных, которые могут отличаться от одного конверта к другому.

Если используется исключающая канонизация, как метод канонизации или преобразования, то пространство имен схем XACML, используемое элементами в объекте данных XACML должно быть связано с префиксами и включено в параметр `InclusiveNamespacesPrefixList` к http://www.w3.org/2001/10/xml-exc-c14n# (смотрите W3C Canonicalization:2002).

11.2.2 Дополнительные соображения по поводу канонизации

Дополнительные преобразования объекта данных XACML обычно должны выполняться, для того чтобы гарантировать сопоставимость подписанного объекта данных и объекта данных, который следует подтвердить. Здесь перечислены некоторые из таких преобразований, но этот профиль не пытается устанавливать алгоритмы для их выполнения.

Если в объект данных XACML включены элементы данных, которые могут быть представлены более, чем в одной форме (такие, как (TRUE, FALSE), (1,0), (true,false)), то должен быть определен и установлен метод преобразования для нормализации этих элементов данных.

В этом профиле рекомендуется применение следующих канонизаций к значениям соответствующих типов данных, независимо от того, появляются ли они в значениях атрибутов XML или в атрибутах XACML.

1) Если каноническое представление для типа данных, определенного в XACML определено в http://www.w3.org/2001/XMLSchema, то значение типа данных должно быть помещено в каноническую форму, установленную в http://www.w3.org/2001/XMLSchema. В нее включены логические {"true", "false"}, удвоения, dateTime, время, дата и hexBinary (прописной).
2) http://www.w3.org/2001/XMLSchema#anyURI – Использовать каноническую форму, определенную в IETF RFC 2396.
3) http://www.w3.org/2001/XMLSchema#base64Binary – Удалить все разрывы строк и пробелы. Удалить все символы, следующие за первой последовательностью символов "=" characters. Преобразование Base64 (идентификатор: http://www.w3.org/TR/xmldsig-core/#sec-Base-64) может быть полезен при выполнении такой канонизации.

4) `urn:oasis:names:tc:xacml:1.0:data-type:x500Name` – Сначала нормализовать в соответствии с IETF RFC 2253. Если в каком-либо RDN содержится множество пар `attributeTypeAndValue`, сделать перестановку `AttributeValuePairs` в этом RDN в восходящем порядке, если сравнение происходит в виде строк октетов (смотрите 11.6/X.690).

5) `urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name` – Нормализовать часть домена имен до строчных букв.

6) выражение XPath – Применить http://www.w3.org/2002/06/xmldsig-filter2, для того чтобы поставить выражение XPath в каноническую форму.

11.3 Подписание схем

Анализ любого объекта данных XACML зависит от наличия точной копии всех схем, от которых зависит этот объект данных XACML. Заметим, что включение URI схемы в атрибуты экземпляра схемы XACML не гарантирует факт использования точных копий схемы: атакующий может подменить схему на поддельную, в которой содержится верный идентификатор. Подписи могут помочь защититься от подмены или изменения схем, от которых зависит объект данных XACML. Использование подписей с этой целью описано в данном пункте.

В большинстве случаев, подписант объекта данных должен включать элемент <Reference> для каждой схемы, от которой зависит объект данных XACML, в элемент <SignedInfo>, содержащий <Reference>, до или включая сам объект данных XACML.

В некоторых случаях подписант объекта данных осведомлен о том, что у всех PDP, которые будут оценивать объект данных XACML, будут иметься точные копии определенных схем, требующихся для анализа этого объекта данных, и он не намерен принуждать PDP подтверждать список сообщений для этих схем. В этих случаях подписант объекта данных может пропускать элементы <Reference> для любых схем, подтверждение которых не требуется.

12 Профиль иерархического ресурса XACML

Часто бывает, что ресурс организован иерархическим образом. Примерами тому являются системы файлов, документы XML и организации. В этом пункте устанавливается, каким образом XACML может предоставить контроль доступа для ресурса, организованного иерархическим образом.

Чем же отличаются ресурсы, организованные иерархическим образом? Прежде всего, стратегии по всем иерархиям часто применяют одни и те же элементы контроля доступа ко всем поддеревьям этой иерархии. Возможность выразить единое ограничение стратегии, которое будет применяться ко всему поддереву узлов в иерархии, а не сталкиваться с необходимостью установления отдельного ограничения для каждого узла, усиливает как простоту применения, так и вероятность того, что эта стратегия будет верно отражать желательные элементы контроля доступа. Другой особенностью иерархических ресурсов является то, что доступ к одному узлу может зависеть от значения другого узла. Например, медицинскому пациенту может быть предоставлен доступ к узлу "диагноз" в медицинской записи документа XML только если имя пациента сопоставимо со значением в узле "имя пациента". Если у нас такой случай, то запрашиваемый узел не может обрабатываться изолированно от остальных узлов в иерархии, а у PDP должен иметься доступ к значениям других узлов. Наконец, идентичность узлов в иерархии часто зависит от позиции узла в иерархии; также может существовать множество способов для описания идентичности единого узла. Для того чтобы стратегии применялись к узлам так, как было намечено, нужно уделить внимание согласованным представлениям для идентичности этих узлов. Иначе, запросчик может обойти элементы контроля доступа, запрашивая узел с использованием идентичности, которая отличается от идентичности, используемой в стратегии.

Ресурс, организованный как иерархия, может быть "деревом" (иерархия с единым корнем) или "лесом" (иерархия с множеством корней), но у иерархии не может быть циклов. Другой термин для этих двух типов иерархии – это "Ориентированный ациклический граф" или "DAG". Все такие ресурсы называются иерархическими ресурсами в этом профиле. Документ XML всегда структурирован, как "дерево". Другие типы иерархических ресурсов, такие как файлы в системе файлов, которые поддерживают компоновки, могут быть структурированы, как "лес".

С узлами в иерархическом ресурсе обращаются, как с отдельными ресурсами. Решение об авторизации, которое разрешает доступ к внутреннему узлу не предполагает разрешение доступа к его узлам-потомкам. Решение об авторизации, которое запрещает доступ к внутреннему узлу, не предполагает запрет доступа к его узлам-потомкам.

Существует три типа средств, установленных в этом профиле, для обращения с иерархическими ресурсами:

- Представление идентичности в узле.
- Запрашивание доступа к узлу.
- Выражение стратегий, которые применимы к одному или более узлов.

Поддержка для каждого из этих средств является необязательной.

В этом пункте обращаются к двум способам представления иерархического ресурса. В первом способе иерархия, частью которой является узел, представлена, как документ XML, включенный в запрос, а запрашиваемый ресурс представлен, как узел в этом документе. Во втором способе запрашиваемый ресурс не представлен, как узел в документе XML, и отсутствует представление иерархии, часть которой включена в запрос. Заметим, что в первом случае нет необходимости в том, чтобы фактический ресурс цели был частью документа XML – он просто представлен таким образом в Запросе. Точно так же, ресурс цели во втором случае может фактически быть частью документа XML, но представляется каким-либо другим способом в Запросе. Таким образом, нет предполагаемой взаимозависимости между структурой ресурса, как представлено в Запросе, и фактической структурой физического ресурса, к которому запрашивается доступ.

Тот факт, что сам документ XML включен в запрос о принятии решения, может оказаться полезным для средств для обращения с ресурсами, представленных как узлы в документах XML. Выражения XPath могут использоваться для ссылки на узлы в этом документе стандартным способом и, могут быть предоставлены уникальные представления для заданных узлов в этом документе. Эти средства недоступны для иерархических ресурсов, которые не представлены, как документы XML. Могут быть предоставлены другие средства в случае таких не-XMLресурсов для установления местоположения запрашиваемого узла в иерархии. В некоторых случаях это может быть сделано с помощью включения позиции узла в иерархии, как части идентичности узла. В других случаях у узла может быть более одной нормативной идентичности, как например, когда путевое имя файла в системе файлов может включать в себя точную ссылку. В таких случаях обработчику контекста PDP XACML может понадобиться доставка идентичностей всех предшественников узла. По всем этим причинам, средства обращения с узлами в документах XML отличаются от средств обращения с узлами в других иерархических ресурсах.

При обращении с иерархическим ресурсом может быть полезно запрашивать решения об авторизации для множества узлов в ресурсе в едином запросе о принятии решения. Можно считать, что этот пункт должен быть иерархически помещен в верхнюю часть профиля Множества ресурсов (смотрите пункт 9), который, в свою очередь, иерархически помещается в верхнюю часть образа действия, установленного в пункте 7. Функциональность в этом пункте может, однако, быть иерархически помещена прямо в функциональность в пункте 7.

В этом пункте для иерархических ресурсов предполагается, что все запросы о доступе к множеству узлов в иерархическом ресурсе были разрешены по отдельным запросам для доступа к единому узлу.

12.1 Представление идентичности узла

Для того чтобы стратегии XACML применялись к узлам в иерархическом ресурсе согласованно, необходимо, чтобы узлы в этом ресурсе были представлены согласованным образом. Если стратегия обращается к узлу, используя одно представление, а запрос обращается к узлу, используя другое представление, то стратегия будет неприменима и безопасность может быть дискредитирована.

В следующих пунктах описываются рекомендуемые представления для узлов в иерархических ресурсах. Альтернативные представления узлов в заданном ресурсе разрешены при условии, что все пункты управления стратегией и все пункты осуществления стратегии, которые имеют дело с этим ресурсом, заключили соглашение об использовании этого альтернативного представления.

12.1.1 Узлы в документах XML

Этот пункт является нормативным, но он необязательный.

Следующий URI должен использоваться в качестве идентификатора для функциональности, установленной в этой части данного профиля:

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-id
```

Идентичностью узла в ресурсе, который представлен, как экземпляр документа XML, должно быть выражение XPath, которое оценивает точно этот один узел в копии ресурса, который содержится в элементе `<ResourceContent>` элемента `<Resource>` элемента `<Request>`.

12.1.2 Узлы в ресурсах, которые не являются документами XML

Этот пункт является нормативным, но он необязательный.

Следующий URI должен использоваться в качестве идентификатора для функциональности, установленной в этой части данного профиля:

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-id
```

Идентичность узла в иерархическом ресурсе, который не представлен, как экземпляр документа XML, должна быть представлена, как URI, который согласуется с IETF RFC 2396. Такие идентификаторы URI имеют следующую форму.

```
<scheme> ":" <authority> "/" <pathname>
```

Ресурсы системы файлов должны использовать схему "file:". Если не установлено ни одной стандартной `<scheme>` для типа ресурса в IETF RFC 2396 или в связанном стандарте для зарегистрированной схемы URI, то URI должен использовать схему "`file`:".

Часть <pathname> идентификатора URI должна иметь следующую форму:

```
<root name> [ "/" <node name> ]*
```

Последовательность значений <root name> и <node name> должна соответствовать отдельным иерархическим именам компонентов предшественников представленного узла по пути от узла <root> к представленному узлу.

Должна использоваться следующая канонизация.

- Кодированием URI должно быть UTF8.
- Независимыми от регистра частями URI должны быть строчные буквы.
- Переход символов должен согласовываться с IETF RFC 2396.
- Часть <authority> URI должна быть установлена и должна быть стандартным представлением органа для заданного типа ресурса. Если <authority> может быть установлена с помощью использования имени системы доменных имен (DNS) или числового IPv4, или адреса IPv6, то должно использоваться имя DNS.
- Компоненты части <pathname> URI должны быть установлены с использованием канонической формы для таких компонентов пути в <authority>.
- В соответствии с IETF RFC 2396, символом-разделителем между иерархическими компонентами части <pathname> URI должен быть символ "/". Последовательности символа "/" должны быть разрешены в единый символ "/". Идентичности узла не должны заканчиваться символом "/".
- <pathname> не должен содержать программных связей.
- Все значения <pathname> должны быть абсолютными.
- Если имеется более одного полностью разрешенного, абсолютного пути от <root> в <authority> к представленному узлу, то отдельный атрибут ресурса с AttributeId "urn:oasis:names:tc:xacml:1.0:resource:resource-id" и DataType http://urn:oasis:names:tc:xacml:1.0:data-type:anyURI должен быть представлен в Контексте запроса для каждого такого пути.

12.2 Запрашивание доступа к узлу

Для того чтобы применение стратегий XACML к узлам в иерархическом ресурсе было согласованным, для каждого контекста запроса, в котором представлен запрос о доступе к узлу в этом ресурсе, необходимо использовать согласованное описание этого доступа к узлу. Если стратегия ссылается на какие-либо ожидаемые атрибуты узла, но в контексте запроса эти атрибуты не содержатся, или если атрибуты не выражены в ожидаемом виде, то стратегия может оказаться неприменимой и безопасность может быть дискредитирована.

В следующих пунктах описываются рекомендуемые описания контекста запроса о доступе к узлам в иерархических ресурсах. Альтернативные представления таких запросов разрешены при условии, что все пункты управления стратегией и все пункты осуществления стратегии, которые имеют дело с этим ресурсом, заключили соглашение об использовании этого альтернативного представления.

12.2.1 Узлы в документе XML

Этот пункт является нормативным, но он необязательный.

Следующий URI должен использоваться в качестве идентификатора для функциональности, установленной в этой части данного профиля:

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-req
```

Атрибуты с AttributeIds со значениями:

```
"urn:oasis::names:tc:xacml:2.0:resource:resource-parent"
"urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor"
```

и:

```
"urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor-or-self"
```

являются необязательными для реализации. Если они поддерживаются для использования в ресурсах, представленных, как документы XML, то следующие URI должны использоваться в качестве идентификаторов для функциональности, которую они представляют:

```
"urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-req:resource-parent"
"urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-req:resource-ancestor"
```

и:

```
"urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-req:resource-ancestor-
or-self"
```

Для того чтобы запросить доступ к ресурсу, представленному, как узел в документе XML, в элементе <Resource> контекста запроса должны содержаться следующие элементы и атрибуты XML.

- Элемент <ResourceContent>, в котором содержится экземпляр всего документа XML, частью которого является запрашиваемый узел.
- Элемент <Attribute> с AttributeId со значением "urn:oasis::names:tc:xacml:1.0:resource:resource-id" и DataType со значением "urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression". <AttributeValue> этого <Attribute> должен быть выражением XPath, узлом контекста которого должен быть один и только один дочерний элемент <ResourceContent>. Это выражение XPath должно оценивать набор узлов, содержащий единый узел в элементе <ResourceContent>, который является узлом, к которому запрашивается доступ. Этот <Attribute> может устанавливать Issuer.
- Элемент <Attribute> с AttributeId со значением "urn:oasis::names:tc:xacml:2.0:resource:resource-parent" и DataType со значением "urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression". <AttributeValue> этого <Attribute> должен быть выражением XPath, узлом контекста которого должен быть один и только один дочерний элемент <ResourceContent>. Это выражение XPath должно оценивать набор узлов, содержащий единый узел в элементе <ResourceContent>, который является непосредственным родителем узла, представленного в атрибуте "resource-id". Этот <Attribute> может устанавливать Issuer.
- Для каждого узла в экземпляре документа XML, который является предшественником узла, представленного атрибутом "resource-id", элемент <Attribute> с AttributeId со значением "urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor" и DataType со значением "urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression". <AttributeValue> этого <Attribute> должен быть выражением XPath; узлом контекста для этого выражения XPath должен быть один и только один дочерний элемент <ResourceContent>. Это выражение XPath должно оценивать набор узлов, содержащий единый узел в элементе <ResourceContent>, который является соответственным предшественником узла, представленного в атрибуте "resource-id". Для каждого атрибута "resource-parent", должен существовать соответствующий атрибут "resource-ancestor". Этот <Attribute> может устанавливать Issuer.
- Для каждого узла в экземпляре документа XML, который является предшественником узла, представленного атрибутом "resource-id", и для самого узла "resource-id", элемент <Attribute> с AttributeId со значением "urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor-or-self" и DataType со значением "urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression". <AttributeValue> этого <Attribute> должен быть выражением XPath; узлом контекста для этого выражения XPath должен быть один и только один дочерний элемент <ResourceContent>. Это выражение XPath должно оценивать набор узлов, содержащий единый узел в элементе <ResourceContent>, который является соответственным предшественником узла, представленного в атрибуте "resource-id" или это сам узел "resource-id". Для каждого атрибута "resource-parent" и "resource-id", должен существовать соответствующий атрибут "resource-ancestor-or-self". Этот <Attribute> может устанавливать Issuer.

В элемент <Resource> могут быть включены дополнительные атрибуты. В частности, может быть включен следующий атрибут.

- Элемент <Attribute> с AttributeId со значением "urn:oasis::names:tc:xacml:2.0:resource:document-id" и DataType со значением "urn:oasis:names:tc:xacml:1.0:data-type:anyURI". <AttributeValue> этого <Attribute> должен быть URI, который идентифицирует документ XML, частью которого является запрашиваемый ресурс, и копия которого представлена в элементе <ResourceContent>. Этот <Attribute> может устанавливать Issuer.

12.2.2 Узлы в ресурсе, который не является документом XML

Этот пункт является нормативным, но он необязательный.

Следующий URI должен использоваться в качестве идентификатора для функциональности, установленной в этой части данного профиля:

urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-req

Атрибуты с AttributeIds со значением:

"urn:oasis::names:tc:xacml:2.0:resource:resource-parent"

"urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor"

и:

"urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor-or-self"

являются необязательными для реализации. Если они поддерживаются для использования в ресурсах, не представленных, как документы XML, то следующие URI должны использоваться в качестве идентификаторов для функциональности, которую они представляют:

"urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-req:resource-parent"

"urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-req:resource-ancestor"

и:

"urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-req:resource-ancestor-or-self"

Для того чтобы запросить доступ к узлу в иерархическом ресурсе, не представленному, как документ XML, в элементе <Resource> контекста запроса не должен содержаться элемент <ResourceContent>. В элементе <Resource> контекста запроса должны содержаться следующие элементы и атрибуты XML. Заметим, что у узла в иерархическом ресурсе, не представленного, как документ XML может быть множество родителей. Например, в системе файлов, которая поддерживает точную компоновку, может быть множество нормативных путей к единому файлу. Каждый такой путь может содержать разные наборы родителей и предшественников.

– Для каждого нормативного представления запрашиваемого узла, элемент <Attribute> с AttributeId со значением "urn:oasis::names:tc:xacml:1.0:resource:resource-id". <AttributeValue> этого <Attribute> должен быть уникальной нормативной идентичностью узла, к которому запрашивается доступ. DataType этого <Attribute> должен зависеть от представления, выбранного для идентичности узлов в этом конкретном ресурсе. Этот <Attribute> может устанавливать Issuer.

– Для каждого непосредственного родителя узла, установленного в атрибуте или атрибутах "resource-id" и для каждого нормативного представления этого родительского узла, элемент <Attribute> с AttributeId "urn:oasis::names:tc:xacml:2.0:resource:resource-parent". <AttributeValue> этого <Attribute> должен быть нормативной идентичностью этого родительского узла. DataType этого <Attribute> должен зависеть от представления, выбранного для идентичности узлов в этом конкретном ресурсе. Этот <Attribute> может устанавливать Issuer. Если запрашиваемый узел является частью леса, а не частью отдельного дерева, или если у родительского узла имеется более одного нормативного представления, то должен существовать, по крайней мере, один экземпляр этого атрибута для каждого родителя вдоль каждого пути к множеству корней, чьим потомком является запрашиваемый узел, и для каждого нормативного представления каждого такого родителя.

– Для каждого предшественника узла, установленного в атрибуте или атрибутах "resource-id", и для каждого нормативного представления этого узла-предшественника элемент <Attribute> с AttributeId "urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor". <AttributeValue> этого <Attribute> должен быть нормативной идентичностью узла-предшественника. DataType этого <Attribute> должен зависеть от представления, выбранного для идентичности узлов в этом конкретном ресурсе. Этот <Attribute> может устанавливать Issuer. Для каждого атрибута "resource-parent" должен существовать соответствующий атрибут "resource-ancestor". Если запрашиваемый узел является частью леса, а не частью отдельного дерева, или если у узла-предшественника имеется более одного нормативного представления, то должен существовать, по крайней мере, один экземпляр этого атрибута для каждого предшественника вдоль каждого пути к множеству корней, чьим потомком является запрашиваемый узел, и для каждого нормативного представления каждого такого предшественника. Порядок расположения значений этого атрибута не обязательно отражает положение каждого узла-предшественника в иерархии.

– Для каждого предшественника узла, установленного в атрибуте или атрибутах "resource-id", и для каждого нормативного представления этого узла-предшественника, и для каждого нормативного представления самого узла "resource-id", элемент <Attribute> с AttributeId "urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor-or-self". <AttributeValue> этого <Attribute> должен быть соответствующей нормативной идентичностью узла-предшественника или самого узла "resource-id". DataType этого <Attribute> должен зависеть от представления, выбранного для идентичности узлов в этом конкретном ресурсе. Этот <Attribute> может устанавливать Issuer. Для каждого атрибута "resource-ancestor" и "resource-id" должен существовать соответствующий атрибут "resource-ancestor-or-self". Если запрашиваемый узел является частью леса, а не частью отдельного дерева, или если у узла-предшественника имеется более одного нормативного представления, то должен существовать, по крайней мере, один экземпляр этого атрибута для каждого предшественника вдоль каждого пути к множеству корней, чьим потомком является

запрашиваемый узел, и для каждого нормативного представления каждого такого предшественника. Порядок расположения значений этого атрибута не обязательно отражает положение каждого узла-предшественника в иерархии.

В элемент <Resource> могут быть включены дополнительные атрибуты.

12.3 Заявление стратегий, которые применяются к узлам

Этот пункт является информативным.

В этом пункте описываются различные способы установления предиката стратегии, который можно применять к множеству узлов в иерархическом ресурсе. Этот список не претендует на всесторонность.

12.3.1 Стратегии, применимые к узлам в любом иерархическом ресурсе

Этот пункт является информативным.

Атрибуты ресурса со следующими значениями AttributeId, описанные в пункте 12.5, могут использоваться для заявления стратегий, которые применяются к одному или более узлов в любом иерархическом ресурсе.

```
urn:oasis:names:tc:xacml:2.0:resource:resource-parent
urn:oasis:names:tc:xacml:2.0:resource:resource-ancestor
urn:oasis:names:tc:xacml:2.0:resource:resource-ancestor-or-self
```

Заметим, что <ResourceAttributeDesignator>, который ссылается на атрибут "resource-parent", "resource-ancestor" или "resource-ancestor-or-self", будет возвращать "мешок" значений, представляющий все нормативные идентичности всех родителей, предшественников или предшественников плюс сам ресурс, соответственно, того ресурса, к которому запрашивается доступ. Представления идентичностей этих родителей, предшественников или самого себя необязательно показывает путь от корня иерархии к соответственному родителю, предшественнику или самому себе, если только не используется представление, рекомендованное в пункте 12.2.2, Узлы в ресурсе, который не является документом XML.

Стандартный "мешок" XACML и функции "мешка" более высокого порядка могут использоваться для заявления стратегий, которые применимы к одному или более узлов в любом иерархическом ресурсе. Узлы, которые используются, как аргументы этих функций, могут быть установлены с использованием <ResourceAttributeDesignator> со значением AttributeId "resource-parent", "resource-ancestor", или "resource-ancestor-or-self".

12.3.2 Стратегии, применимые только к узлам в документах XML

Этот пункт является информативным.

Для иерархических ресурсов, которые представлены, как экземпляры документа XML, может использоваться следующая функция для заявления о предикатах стратегии, которые применимы к одному или более узлам в этом ресурсе.

```
urn:oasis:names:tc:xacml:2.0:function:xpath-node-match
```

Стандартный элемент <AttributeSelector> может использоваться в стратегиях для ссылок на весь ресурс или части ресурса, представленные, как документ XML, и заключенные в элемент <ResourceContent> контекста запроса.

Стандартный "мешок" XACML и функции "мешка" более высокого порядка могут использоваться для заявления стратегий, которые применимы к одному или более узлам в ресурсе, представленном, как документ XML. Узлы, которые используются, как аргументы этих функций, могут быть установлены с использованием <AttributeSelector>, который выбирает часть элемента <ResourceContent> элемента <Resource>.

12.3.3 Стратегии, применимые только к узлам в не-XML ресурсах

Этот пункт является информативным.

Для иерархических ресурсов, которые не представлены, как экземпляры документа XML и, если используется представление URI узлов, как установлено в этом профиле, могут использоваться следующие функции для заявления стратегий, которые применимы к одному или более узлам в этом ресурсе.

```
urn:oasis:names:tc:xacml:1.0:function:anyURI-equal
urn:oasis:names:tc:xacml:1.0:function:regexp-uri-match
```

12.4 Новый DataType: xpath-expression

Этот пункт является нормативным, но не обязательным.

Следующее значение для значения атрибута DataType XML может поддерживаться для использования с иерархическими ресурсами, представленными, как документы XML. Поддержка для этого DataType требуется, для того чтобы поддержать пункт 12.1.1.

DataType, представленный следующими URI, представляет выражение XPath. Значения атрибутов, у которых есть этот DataType, должны быть строками, которые нужно интерпретировать, как выражения XPath. Результатом оценки такого атрибута должен быть набор узлов, который появляется в результате оценки выражения XPath. Если строка не является действующим выражением XPath, то результатом оценки этого атрибута должно быть "Indeterminate".

```
urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression
```

12.5 Идентификаторы нового атрибута

Этот пункт является нормативным, но не обязательным.

12.5.1 document-id

Следующий идентификатор показывает идентичность документа XML, который представляет иерархию, частью которой является запрашиваемый ресурс и, копия которого присутствует в элементе <ResourceContent>. Когда бы ни запрашивался доступ к узлу в ресурсе, представленном, как документ XML, один или более экземпляров атрибута с этим AttributeId может быть предоставлен в элементе <Resource> контекста запроса. DataType этих атрибутов должен быть "urn:oasis:names:tc:xacml:1.0:data-type:anyURI".

```
urn:oasis:names:tc:xacml:2.0:resource:document-id
```

12.5.2 resource-parent

Следующий идентификатор показывает одну нормативную идентичность одного родительского узла в дереве или лесу, частью которого является запрашиваемый узел. Когда бы ни запрашивался доступ к узлу в иерархическом ресурсе, один или более экземпляров атрибута с этим AttributeId может быть предоставлен в элементе <Resource> контекста запроса для каждого нормативного представления каждого узла, который является родителем запрашиваемого узла.

```
urn:oasis:names:tc:xacml:2.0:resource:resource-parent
```

12.5.3 resource-ancestor

Следующий идентификатор показывает одну нормативную идентичность одного узла-предшественника в дереве или лесу, частью которого является запрашиваемый узел. Когда бы ни запрашивался доступ к узлу в иерархическом ресурсе, один экземпляр атрибута с этим AttributeId может быть предоставлен в элементе <Resource> контекста запроса для каждого нормативного представления каждого узла, который является предшественником запрашиваемого узла.

```
urn:oasis:names:tc:xacml:2.0:resource:resource-ancestor
```

12.5.4 resource-ancestor-or-self

Следующий идентификатор показывает одну нормативную идентичность одного узла-предшественника в дереве или лесу, частью которого является запрашиваемый узел или одну нормативную идентичность самого запрашиваемого узла. Когда бы ни запрашивался доступ к узлу в иерархическом ресурсе, один экземпляр атрибута с этим AttributeId может быть предоставлен в элементе <Resource> контекста запроса для каждого нормативного представления каждого узла, который является предшественником запрашиваемого узла и для каждого нормативного представления самого запрашиваемого узла.

```
urn:oasis:names:tc:xacml:2.0:resource:resource-ancestor-or-self
```

12.6 Идентификаторы нового профиля

Следующие значения URI должны использоваться, как идентификаторы для функциональности, установленной в разных пунктах этого профиля:

Пункт 12.1.1: Узлы в документах XML

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-id
```

Пункт 12.1.2: Узлы в ресурсах, которые не являются документами XML

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-id
```

Пункт 12.2.1: Узлы в документах XML

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-req
```

Поддержка для атрибутов "resource-parent", "resource-ancestor", и "resource-ancestor-or-self" является необязательной внутри этого пункта, поэтому у них отдельные идентификаторы:

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-req:resource-parent
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-req:resource-ancestor
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-req:resource-ancestor-
or-self
```

Пункт 12.2.2: Узлы в ресурсе, который не является документом XML

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-req
```

Поддержка для атрибутов "`resource-parent`", "`resource-ancestor`", и "`resource-ancestor-or-self`" является необязательной внутри этого пункта, поэтому у них отдельные идентификаторы:

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-req:resource-parent
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-req:resource-
ancestor
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-req:resource-
ancestor-or-self
```

13 Профиль стратегии секретности

Два обязательства, накладываемых на хранителя данных, должны гарантировать, что использование личных данных ограничено исполнением тех целей, для которых они собраны, или для других целей, которые не являются несовместимыми с этими, и предотвратить разглашение личных данных, за исключением случаев, когда субъект дает свое согласие или по требованию правоохранительных органов. В этом пункте предоставлен профиль для стандартных атрибутов и стандартный элемент `<Rule>` для осуществления этих обязательств, связанных с целью, для которой лично идентифицируемая информация собирается и используется.

13.1 Стандартные атрибуты

В этом профиле определено два атрибута.

```
urn:oasis:names:tc:xacml:2.0:resource:purpose
```

Этот атрибут, типа "http://www.w3.org/2001/XMLSchema#string", показывает цель, для которой был собран ресурс данных. Владелец ресурса должен быть информирован и должен дать свое согласие на использование данного ресурса для этой цели. Значением атрибута может быть регулярное выражение. Стратегия секретности хранителя должна определять семантику всех возможных значений.

```
urn:oasis:names:tc:xacml:2.0:action:purpose
```

Этот атрибут, типа "http://www.w3.org/2001/XMLSchema#string", показывает цель, для которой запрашивается доступ к ресурсу данных. Цели действия могут быть организованы иерархическим образом, в этом случае значение должно представлять узел в иерархии.

13.2 Стандартные правила: Цель сопоставления

Это правило должно использоваться с алгоритмом объединения правил "`urn:oasis:names:tc:xacml:2.0:rule-combining-algorithm:deny-overrides`". Оно ставит следующее условие: доступ должен быть запрещен, если цель, для которой запрашивается доступ, с помощью регулярного выражения сопоставления, не сопоставима с целью, для которой был собран этот ресурс данных.

```
<?xml version="1.0" encoding="UTF-8"?>
<Rule xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os" RuleId="
urn:oasis:names:tc:xacml:2.0:matching-purpose"
Effect="Permit">
      <Condition FunctionId="urn:oasis:names:tc:xacml:2.0:function:regexp-
string-match">
            <ResourceAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:2.0:resource:purpose"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
            <ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:2.0:action:purpose"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
      </Condition>
</Rule>
```

Приложение A

Типы данных и функции

A.1 Введение

В этом приложении устанавливаются типы данных и функции, используемые в XACML для создания предикат для условий и сопоставлений цели.

В данной Рекомендации описываются простые типы данных и "мешки". Даны имена стандартным функциям и описана их операционная семантика.

ПРИМЕЧАНИЕ. – Смотрите [IEEE 754] и [RBAC] для строкового представления информации численных значений

A.2 Типы данных

Хотя экземпляры XML представляют все типы данных в виде строк, PDP XACML должен обосновать типы данных, потому что, несмотря на то, что у них имеется строковое представление, но они не просто строки. Типы, такие как строчные, логические, целочисленные и чисел с двойной точностью должны быть преобразованы из их строкового представления XML в значения, которые можно сравнивать со значениями в их домене языкового общения, таком как числа. Следующие простые типы данных установлены для использования XACML и имеют явные представления данных:

- http://www.w3.org/2001/XMLSchema#string
- http://www.w3.org/2001/XMLSchema#boolean
- http://www.w3.org/2001/XMLSchema#integer
- http://www.w3.org/2001/XMLSchema#double
- http://www.w3.org/2001/XMLSchema#time
- http://www.w3.org/2001/XMLSchema#date
- http://www.w3.org/2001/XMLSchema#dateTime
- http://www.w3.org/2001/XMLSchema#anyURI
- http://www.w3.org/2001/XMLSchema#hexBinary
- http://www.w3.org/2001/XMLSchema#base64Binary
- urn:oasis:names:tc:xacml:2.0:data-type:dayTimeDuration
- urn:oasis:names:tc:xacml:2.0:data-type:yearMonthDuration
- urn:oasis:names:tc:xacml:1.0:data-type:x500Name
- urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name
- urn:oasis:names:tc:xacml:2.0:data-type:ipAddress
- urn:oasis:names:tc:xacml:2.0:data-type:dnsName

Для улучшения функциональной совместимости рекомендуется, чтобы все временные ссылки проводились во времени UTC (Всеобщее скоординированное время).

PDP XACML должен быть способен преобразовывать строковые представления в различные простые типы данных.

ПРИМЕЧАНИЕ. – Для целых чисел и чисел с двойной точностью, XACML должен использовать преобразования, описанные в [IEEE 754].

В XACML определено шесть типов данных; это:

```
"urn:oasis:names:tc:xacml:2.0:data-type:dayTimeDuration"
"urn:oasis:names:tc:xacml:2.0:data-type:yearMonthDuration"
"urn:oasis:names:tc:xacml:1.0:data-type:x500Name"
"urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name"
"urn:oasis:names:tc:xacml:2.0:data-type:ipAddress"
"urn:oasis:names:tc:xacml:2.0:data-type:dnsName"
```

Эти типы представляют идентификаторы для субъектов или ресурсов и появляются в нескольких стандартных приложениях, таких как TLS/SSL и электронная почта.

A.2.1 Длительность в днях и в разах

Простой тип данных "`urn:oasis:names:tc:xacml:2.0:data-type:dayTimeDuration`" определяется, как ограничение типа **xs:duration**, сохраняющее только компоненты знака, года и месяца.

```
<xs:simpleType name='urn:oasis:names:tc:xacml:2.0:data-
type:yearMonthDuration'>
  <xs:restriction base='xs:duration'>
    <xsd:pattern value="[-]?P\p{Nd}+(Y(\p{Nd}+M)?|M)"/>
  </xs:restriction>
</xs:simpleType>
```

Значение `yearMonthDuration` дается в размерности месяцев и представляет собой:

```
('value of the year component' * 12) + ('value of the month component')
```

Если знак у компонента "-", то результирующее значение отрицательное; иначе, результирующее значение положительное.

A.2.2 Длительность в годах и месяцах

Простой тип данных "`urn:oasis:names:tc:xacml:2.0:data-type:yearMonthDuration`" определяется, как ограничение типа **xs:duration**, сохраняющее только компоненты знака, дня, часа, минуты и секунды.

```
<xs:simpleType name='urn:oasis:names:tc:xacml:2.0:data-
type:dayTimeDuration'>
  <xs:restriction base='xs:duration'>
     <xsd:pattern value="[-
]?P(\p{Nd}D(T(\p{Nd}+(H(\p{Nd}+(M(\p{Nd}+(\.\p{Nd}*)?S
      |\.\p{Nd}+S)?|(\.\p{Nd}*)?S)|(\.\p{Nd}*)?S)?|M(\p{Nd}+
      (\.\p{Nd}*)?S|\.\p{Nd}+S)?|(\.\p{Nd}*)?S)|\.\p{Nd}+S))?
      |T(\p{Nd}+(H(\p{Nd}+(M(\p{Nd}+(\.\p{Nd}*)?S|\.\p{Nd}+S)?
      |(\.\p{Nd}*)?S)|(\.\p{Nd}*)?S)?|M(\p{Nd}+(\.\p{Nd}*)?S|\.\p{Nd}+S)?
      |(\.\p{Nd}*)?S)|\.\p{Nd}+S))"/>
  </xs:restriction>
</xs:simpleType>
```

Значение `dayTimeDuration` дается в размерности секунд и представляет собой:

```
('value of the day component' * 24) +
('value of the hour component' * 60) +
('value of the minute component' * 60) +
('value of the second component')
```

Если знак у компонента "-", то результирующее значение отрицательное; иначе, результирующее значение положительное.

A.2.3 Имя справочника X.500

Простой тип данных "`urn:oasis:names:tc:xacml:1.0:data-type:x500Name`" представляет Выделенные имена X.520. Действительный синтаксис для такого имени описан в IETF RFC 2253.

A.2.4 IETF RFC 822 name

Простой тип данных "`urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name`" представляет адрес электронной почты. Действительный синтаксис для такого имени описан в IETF RFC 2821, 4.1.2.

A.2.5 Адрес IP

Простой тип данных "`urn:oasis:names:tc:xacml:2.0:data-type:ipAddress`" представляет адрес сети IPv4 или IPv6 с необязательной маской и необязательным портом или цепью портов. Синтаксис должен быть:

```
ipAddress = address [ "/" mask ] [ ":" [ portrange ] ]
```

Для адреса IPv4 адрес и маска форматируются в соответствии с синтаксисом для "host" в IETF RFC 2396, 3.2.

Для адреса IPv6 адрес и маска форматируются в соответствии с синтаксисом для "ipv6reference" в IETF RFC 2732. (Заметим, что адрес или маска IPv6 в этом синтаксисе включается в литеральные скобки "[" "]".)

A.2.6 Имя DNS

Простой тип данных "`urn:oasis:names:tc:xacml:2.0:data-type:dnsName`" представляет имя главного компьютера Системы доменных имен (DNS) с необязательным портом или цепью портов. Синтаксис должен быть:

```
dnsName = hostname [ ":" portrange ]
```

Имя главного компьютера форматируется в соответствии с IETF RFC 2396, 3.2, за исключением случаев, когда групповой символ "*" может использоваться в самом левом компоненте имени главного компьютера, чтобы показать "любой поддомен" под доменом, установленным справа от него.

Как для "urn:oasis:names:tc:xacml:2.0:data-type:ipAddress", так и "urn:oasis:names:tc:xacml:2.0:data-type:dnsName" типов данных синтаксис для порта или цепи портов должен быть:

```
portrange = portnumber | "-"portnumber | portnumber"-"[portnumber]
```

где "portnumber" это номер порта в десятичном выражении. Если номер порта в форме "-x", где "x" – это номер порта, то цепью являются все порты, пронумерованные "x" и ниже. Если номер порта в форме "x-", то цепью являются все порты, пронумерованные "x" и выше.

A.3 Функции

В XACML устанавливаются следующие функции. Если аргумент одной из этих функций был оценен, как "Indeterminate", то эта функция должна быть установлена в "Indeterminate".

A.3.1 Предикаты равенства

Следующие функции являются функциями равенства для различных простых типов данных. Каждая функция для конкретного типа данных следует установленной стандартной договоренности для этого типа данных.

```
urn:oasis:names:tc:xacml:1.0:function:string-equal
```

Эта функция должна принимать два аргумента типа данных "http://www.w3.org/2001/XMLSchema#string" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Функция должна возвращать "True" если, и только если значение обоих этих аргументов равной длины и определено, что каждая строка равна побайтно в соответствии с функцией "integer-equal". Иначе, она должна возвращать "False".

```
urn:oasis:names:tc:xacml:1.0:function:boolean-equal
```

Эта функция должна принимать два аргумента типа данных "http://www.w3.org/2001/XMLSchema#boolean" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Функция должна возвращать "True" если, и только если эти аргументы равны. Иначе, она должна возвращать "False".

```
urn:oasis:names:tc:xacml:1.0:function:integer-equal
```

Эта функция должна принимать два аргумента типа данных "http://www.w3.org/2001/XMLSchema#integer" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean".

ПРИМЕЧАНИЕ 1. – Смотрите [IEEE 754] для получения информации о целочисленной оценке целых чисел.

```
urn:oasis:names:tc:xacml:1.0:function:double-equal
```

Эта функция должна принимать два аргумента типа данных "http://www.w3.org/2001/XMLSchema#double" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean".

ПРИМЕЧАНИЕ 2. – Смотрите [IEEE 754] о том, как оценивать числа с двойной точностью.

```
urn:oasis:names:tc:xacml:1.0:function:date-equal
```

Эта функция должна принимать два аргумента типа данных "http://www.w3.org/2001/XMLSchema#date" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Функция должна возвращать "True" если, и только если значения этих двух аргументов равны. Если у какого-либо аргумента не хватает явной информации о временной зоне, то значение, задающее временную зону должно быть предоставлено реализацией.

```
urn:oasis:names:tc:xacml:1.0:function:time-equal
```

Эта функция должна принимать два аргумента типа данных "http://www.w3.org/2001/XMLSchema#time" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Функция должна возвращать "True" если, и только если значения этих двух аргументов равны. Если у какого-либо аргумента не хватает явной информации о временной зоне, то значение, задающее временную зону должно быть предоставлено реализацией.

```
urn:oasis:names:tc:xacml:1.0:function:dateTime-equal
```

Эта функция должна принимать два аргумента типа данных "http://www.w3.org/2001/XMLSchema#dateTime" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Функция должна возвращать "True" если, и только если значения этих двух аргументов равны. Если у какого-либо аргумента не хватает явной информации о временной зоне, то значение, задающее временную зону должно быть предоставлено реализацией.

```
urn:oasis:names:tc:xacml:1.0:function:dayTimeDuration-equal
```

Эта функция должна принимать два аргумента типа данных "urn:oasis:names:tc:xacml:2.0:data-types:dayTimeDuration" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Функция

должна возвращать "True" если, и только если значения этих двух аргументов равны. Заметим, что лексическое представление каждого аргумента должно быть преобразовано в значение, выраженное в дробных секундах.

`urn:oasis:names:tc:xacml:1.0:function:yearMonthDuration-equal`

Эта функция должна принимать два аргумента типа данных "`urn:oasis:names:tc:xacml:2.0:data-types:yearMonthDuration`" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Функция должна возвращать "True" если, и только если, значения этих двух аргументов равны. Заметим, что лексическое представление каждого аргумента должно быть преобразовано в значение, выраженное в целых числах месяцев.

`urn:oasis:names:tc:xacml:1.0:function:anyURI-equal`

Эта функция должна принимать два аргумента типа данных "http://www.w3.org/2001/XMLSchema#anyURI" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Функция должна возвращать "True" если, и только если значения этих двух аргументов равны на основе равенства кодовых точек.

`urn:oasis:names:tc:xacml:1.0:function:x500Name-equal`

Эта функция должна принимать два аргумента типа данных "`urn:oasis:names:tc:xacml:1.0:data-type:x500Name`" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Она должна возвращать "True" если, и только если каждое относительное выделенное имя (RDN) в этих двух аргументах сопоставимо друг с другом. Иначе, она должна возвращать "False". Два имени RDN должны быть признаны сопоставимыми, если и только если результатом следующих операций является "True".

1) Нормализовать эти два аргумента в соответствии с IETF RFC 2253.
2) Если в любом RDN содержится множество пар `attributeTypeAndValue`, то нужно расположить `AttributeValuePairs` в этом RDN в восходящем порядке, когда происходит сравнение строк октетов (описано в пункте 11.6/X.690).
3) Сравнить имена RDN, используя правила в IETF RFC 3280, 4.1.2.4.

`urn:oasis:names:tc:xacml:1.0:function:rfc822Name-equal`

Эта функция должна принимать два аргумента типа данных "`urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name`" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Она должна возвращать "True" если, и только если эти два аргумента равны. Иначе, она должна возвращать "False". IETF RFC 822 name состоит из местной части, за которой следует"@", а затем следует часть домена. Местная часть зависит от конкретных условий, в то время как часть домена (которая обычно является именем главного компьютера DNS) не является зависимой от конкретных условий. Выполнить следующие операции:

1) Нормализовать часть домена каждого аргумента до строчных букв.
2) Сравнить выражения, применяя функцию "`urn:oasis:names:tc:xacml:1.0:function:string-equal`" к нормализованным аргументам.

`urn:oasis:names:tc:xacml:1.0:function:hexBinary-equal`

Эта функция должна принимать два аргумента типа данных "http://www.w3.org/2001/XMLSchema#hexBinary" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Она должна возвращать "True", если последовательность октетов, представленная значением обоих аргументов, имеет равную длину и они равны при конъюнктивном поточечном сравнении с использованием функции "`urn:oasis:names:tc:xacml:1.0:function:integer-equal`". Иначе, она должна возвращать "False". Преобразование из строкового представления в последовательность октетов должна осуществляться так, как установлено в W3C Datatypes:2001, 3.2.15.

`urn:oasis:names:tc:xacml:1.0:function:base64Binary-equal`

Эта функция должна принимать два аргумента типа данных "http://www.w3.org/2001/XMLSchema#base64Binary" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Она должна возвращать "True", если последовательности октетов, представленных значением обоих аргументов, имеют равную длину и они равны при конъюнктивном поточечном сравнении с использованием функции "`urn:oasis:names:tc:xacml:1.0:function:integer-equal`". Иначе, она должна возвращать "False". Преобразование из строкового представления в последовательность октетов должна осуществляться так, как установлено в W3C Datatypes:2001, 3.2.16.

A.3.2 Арифметические функции

Все следующие функции должны принимать аргументы установленного типа данных, целочисленных или чисел удвоенной точностью, и должны возвращать элемент целочисленного и чисел с удвоенной точностью типа данных, соответственно. Однако функции "add" могут принимать более двух аргументов. В любом выражении, содержащем любую из этих функций, если любой аргумент является "Indeterminate", то выражение должно оцениваться, как "Indeterminate". В случае функций деления, делитель равен нулю, то функция должна оцениваться, как"Indeterminate".

ПРИМЕЧАНИЕ. – Каждая оценка функции должна проводиться, как установлено их логическими аналогами в [IEEE 754].

```
urn:oasis:names:tc:xacml:1.0:function:integer-add
```

У этой функции может быть два или более аргументов.

```
urn:oasis:names:tc:xacml:1.0:function:double-add
```

У этой функции может быть два или более аргументов.

```
urn:oasis:names:tc:xacml:1.0:function:integer-subtract
urn:oasis:names:tc:xacml:1.0:function:double-subtract
urn:oasis:names:tc:xacml:1.0:function:integer-multiply
urn:oasis:names:tc:xacml:1.0:function:double-multiply
urn:oasis:names:tc:xacml:1.0:function:integer-divide
urn:oasis:names:tc:xacml:1.0:function:double-divide
urn:oasis:names:tc:xacml:1.0:function:integer-mod
```

Следующие функции должны принимать единый аргумент установленного типа данных. Функции округления и функция "floor" должны принимать единый аргумент типа данных "http://www.w3.org/2001/XMLSchema#double" и возвращать значение типа данных "http://www.w3.org/2001/XMLSchema#double".

```
urn:oasis:names:tc:xacml:1.0:function:integer-abs
urn:oasis:names:tc:xacml:1.0:function:double-abs
urn:oasis:names:tc:xacml:1.0:function:round
urn:oasis:names:tc:xacml:1.0:function:floor
```

A.3.3 Функции преобразования строк

Следующие функции производят преобразования между значениями типа данных "http://www.w3.org/2001/XMLSchema#string" простых типов данных.

```
urn:oasis:names:tc:xacml:1.0:function:string-normalize-space
```

Эта функция должна принимать один аргумент типа данных "http://www.w3.org/2001/XMLSchema#string" и должна нормализовывать это значение с помощью удаления всех символов пробелов, находящихся впереди и сзади.

```
urn:oasis:names:tc:xacml:1.0:function:string-normalize-to-lower-case
```

Эта функция должна принимать один аргумент типа данных "http://www.w3.org/2001/XMLSchema#string" и должна нормализовывать это значение с помощью преобразования каждого прописного символа в его строчный эквивалент.

A.3.4 Функции преобразования численного типа данных

Следующие функции проводят преобразование между типом данных "http://www.w3.org/2001/XMLSchema#integer" и "http://www.w3.org/2001/XMLSchema#double" простых типов данных.

```
urn:oasis:names:tc:xacml:1.0:function:double-to-integer
```

Эта функция должна принимать один аргумент типа данных "http://www.w3.org/2001/XMLSchema#double" и должна округлять свое численное значение до целого числа и возвращать элемент типа данных "http://www.w3.org/2001/XMLSchema#integer".

```
urn:oasis:names:tc:xacml:1.0:function:integer-to-double
```

Эта функция должна принимать один аргумент типа данных "http://www.w3.org/2001/XMLSchema#integer" и должна передавать свое значение элементу типа данных "http://www.w3.org/2001/XMLSchema#double" с тем же численным значением.

A.3.5 Логические функции

В этом пункте содержится спецификация для логических функций, которые оперируют с аргументами типа данных "http://www.w3.org/2001/XMLSchema#boolean".

```
urn:oasis:names:tc:xacml:1.0:function:or
```

Эта функция должна возвращать "False", если у нее нет аргументов, и должна возвращать "True", если по крайней мере один из ее аргументов оценивается как "True". Порядок оценки должен быть установлен от первого аргумента к последнему. Оценка должна остановиться с получением результата "True", если любой из аргументов оценивается как "True", оставляя оставшиеся аргументы без оценки.

```
urn:oasis:names:tc:xacml:1.0:function:and
```

Эта функция должна возвращать "True", если у нее нет аргументов, и должна возвращать "False", если один из ее аргументов оценивается как "False". Порядок оценки должен быть установлен от первого аргумента к последнему. Оценка должна остановиться с получением результата "False", если любой из аргументов оценивается как"False", оставляя оставшиеся аргументы без оценки.

```
urn:oasis:names:tc:xacml:1.0:function:n-of
```

Первым аргументом этой функции должен быть тип данных http://www.w3.org/2001/XMLSchema#integer. Оставшиеся аргументы должны быть типа данных http://www.w3.org/2001/XMLSchema#boolean. Первый аргумент устанавливает минимальное количество оставшихся аргументов, которые должны быть оценены в "True", для того чтобы выражение считалось равным "True".Если первый аргумент равен 0, то результатом должно быть "True". Если количество аргументов после первого меньше, чем значение первого аргумента, то выражение должно в результате быть "Indeterminate". Порядок оценки должен быть таким: сначала оцениваются целочисленные значения, затем оценивают каждый последующий аргумент. Оценка должна остановиться и возвратить "True", если установленное количество аргументов оценено, как "True". Оценка аргументов должна остановиться, если установлено, что оставшиеся аргументы не будут отвечать этому требованию.

```
urn:oasis:names:tc:xacml:1.0:function:not
```

Эта функция должна принимать один аргумент типа данных "http://www.w3.org/2001/XMLSchema#boolean". Если аргумент оценивается как "True", то результатом выражения должно быть "False". Если аргумент оценивается как "False", то результатом выражения должно быть "True".

ПРИМЕЧАНИЕ. – При оценке and, or, или n-of, возможно нет необходимости проводить полную оценку каждого аргумента, для того чтобы выяснить, не появится в результате оценка "Indeterminate". Анализ аргумента относительно работоспособности его атрибутов или другой вид анализа, относительно ошибок, таких, как "деление на ноль", может предоставить аргумент без ошибок. Такое появление аргументов в выражении на позиции после оценки, является указанием на отсутствие необходимости дальнейшей обработки.

A.3.6 Функции численного сравнения

Эти функции формируют минимальный набор для сравнения двух чисел, дающего логический результат.

ПРИМЕЧАНИЕ. – Эти функции должны подчиняться правилам, установленным в [IEEE 754].

```
urn:oasis:names:tc:xacml:1.0:function:integer-greater-than
urn:oasis:names:tc:xacml:1.0:function:integer-greater-than-or-equal
urn:oasis:names:tc:xacml:1.0:function:integer-less-than
urn:oasis:names:tc:xacml:1.0:function:integer-less-than-or-equal
urn:oasis:names:tc:xacml:1.0:function:double-greater-than
urn:oasis:names:tc:xacml:1.0:function:double-greater-than-or-equal
urn:oasis:names:tc:xacml:1.0:function:double-less-than
urn:oasis:names:tc:xacml:1.0:function:double-less-than-or-equal
```

A.3.7 Арифметические функции даты и времени

Эти функции выполняют арифметические операции с датами и временем.

```
urn:oasis:names:tc:xacml:1.0:function:dateTime-add-dayTimeDuration
```

Эта функция должна принимать два аргумента: первый должен быть типом данных "http://www.w3.org/2001/XMLSchema#dateTime", а второй должен быть типом данных "`urn:oasis:names:tc:xacml:2.0:data-types:dayTimeDuration`". Она должна возвращать значение "http://www.w3.org/2001/XMLSchema#dateTime". Эта функция должна возвращать значение прибавлением второго аргумента к первому аргументу в соответствии с W3C DataTypes:2001, Приложение E.

```
urn:oasis:names:tc:xacml:1.0:function:dateTime-add-yearMonthDuration
```

Эта функция должна принимать два аргумента: первый должен быть "http://www.w3.org/2001/XMLSchema#dateTime" а второй должен быть "`urn:oasis:names:tc:xacml:2.0:data-types:yearMonthDuration`". Она должна возвращать результат "http://www.w3.org/2001/XMLSchema#dateTime". Эта функция должна возвращать значение прибавлением второго аргумента к первому аргументу в соответствии с W3C DataTypes:2001, Приложение E.

```
urn:oasis:names:tc:xacml:1.0:function:dateTime-subtract-dayTimeDuration
```

Эта функция должна принимать два аргумента: первый должен быть "http://www.w3.org/2001/XMLSchema#dateTime", а второй должен быть "`urn:oasis:names:tc:xacml:2.0:data-types:dayTimeDuration`". Она должна возвращать результат "http://www.w3.org/2001/XMLSchema#dateTime". Если второй аргумент является положительным интервалом, то эта функция должна возвращать значение прибавлением соответствующего отрицательного интервала, согласно W3C DataTypes:2001, Приложение E. Если второй аргумент является отрицательным интервалом, то результат должен быть таким, как если бы функция "`urn:oasis:names:tc:xacml:1.0:function:dateTime-add-dayTimeDuration`" была приложена к соответствующему положительному интервалу.

urn:oasis:names:tc:xacml:1.0:function:dateTime-subtract-yearMonthDuration

Эта функция должна принимать два аргумента: первый должен быть "http://www.w3.org/2001/XMLSchema#dateTime" а второй должен быть "urn:oasis:names:tc:xacml:2.0:data-types:yearMonthDuration". Она должна возвращать результат "http://www.w3.org/2001/XMLSchema#dateTime". Если второй аргумент является положительным интервалом, то эта функция должна возвращать значение прибавлением соответствующего отрицательного интервала, согласно W3C DataTypes:2001, Приложение E. Если второй аргумент является отрицательным интервалом, то результат должен быть таким, как если бы функция "urn:oasis:names:tc:xacml:1.0:function:dateTime-add-yearMonthDuration" была приложена к соответствующему положительному интервалу.

urn:oasis:names:tc:xacml:1.0:function:date-add-yearMonthDuration

Эта функция должна принимать два аргумента: первый должен быть "http://www.w3.org/2001/XMLSchema#date" а второй должен быть "urn:oasis:names:tc:xacml:2.0:data-types:yearMonthDuration". Она должна возвращать результат "http://www.w3.org/2001/XMLSchema#date". Эта функция должна возвращать значение прибавлением второго аргумента к первому аргументу в соответствии с W3C DataTypes:2001, Приложение E.

urn:oasis:names:tc:xacml:1.0:function:date-subtract-yearMonthDuration

Эта функция должна принимать два аргумента: первый должен быть "http://www.w3.org/2001/XMLSchema#date" а второй должен быть "urn:oasis:names:tc:xacml:2.0:data-types:yearMonthDuration". Она должна возвращать результат "http://www.w3.org/2001/XMLSchema#date". Если второй аргумент является положительным интервалом, то эта функция должна возвращать значение прибавлением соответствующего отрицательного интервала, согласно W3C DataTypes:2001, Приложение E. Если второй аргумент является отрицательным интервалом, то результат должен быть таким, как если бы функция "urn:oasis:names:tc:xacml:1.0:function:date-add-yearMonthDuration" была приложена к соответствующему положительному интервалу.

A.3.8 Функции не численного сравнения

Эти функции выполняют операции сравнения двух аргументов не численных типов.

urn:oasis:names:tc:xacml:1.0:function:string-greater-than

Эта функция должна принимать два аргумента типа данных "http://www.w3.org/2001/XMLSchema#string" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Она должна возвращать "True", если и только если аргументы сравниваются побайтно и, после первоначальных префиксов соответствующих байтов от обоих аргументов, которые считаются равными с помощью "urn:oasis:names:tc:xacml:1.0:function:integer-equal", последующее побайтовое сравнение таково, что байт из первого аргумента больше, чем байт из второго аргумента, с использованием функции "urn:oasis:names:tc:xacml:2.0:function:integer-greater-then". Иначе, она должна возвращать "False".

urn:oasis:names:tc:xacml:1.0:function:string-greater-than-or-equal

Эта функция должна принимать два аргумента типа данных "http://www.w3.org/2001/XMLSchema#string" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Она должна возвращать результат, как если бы оценка производилась с помощью логической функции "urn:oasis:names:tc:xacml:1.0:function:or" с двумя аргументами, содержащими функции "urn:oasis:names:tc:xacml:1.0:function:string-greater-than" и "urn:oasis:names:tc:xacml:1.0:function:string-equal", содержащими исходные аргументы.

urn:oasis:names:tc:xacml:1.0:function:string-less-than

Эта функция должна принимать два аргумента типа данных "http://www.w3.org/2001/XMLSchema#string" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Она должна возвращать "True", если и только если аргументы сравниваются побайтно и, после первоначальных префиксов соответствующих байтов от обоих аргументов, которые считаются равными с помощью "urn:oasis:names:tc:xacml:1.0:function:integer-equal", последующее побайтовое сравнение таково, что байт из первого аргумента меньше, чем байт из второго аргумента, с использованием функции "urn:oasis:names:tc:xacml:1.0:function:integer-less-than". Иначе, она должна возвращать "False".

urn:oasis:names:tc:xacml:1.0:function:string-less-than-or-equal

Эта функция должна принимать два аргумента типа данных "http://www.w3.org/2001/XMLSchema#string" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Она должна возвращать результат, как если бы оценка производилась с помощью функции "urn:oasis:names:tc:xacml:1.0:function:or" с двумя аргументами, содержащими функции "urn:oasis:names:tc:xacml:1.0:function:string-less-than" и "urn:oasis:names:tc:xacml:1.0:function:string-equal", содержащими исходные аргументы.

urn:oasis:names:tc:xacml:1.0:function:time-greater-than

Эта функция должна принимать два аргумента типа данных "http://www.w3.org/2001/XMLSchema#time" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Она должна возвращать "True", если и только если первый аргумент больше, чем второй аргумент в соответствии с отношением упорядочивания, установленном для http://www.w3.org/2001/XMLSchema#time (W3C Signature:2002, 3.2.8). Иначе, она должна возвращать "False".

ПРИМЕЧАНИЕ 1. – Считается, что нельзя сравнивать время, в которое включено значение временной зоны с тем временем, у которого такого значения нет. В таких случаях нужно использовать функцию время-в-диапазоне.

`urn:oasis:names:tc:xacml:1.0:function:time-greater-than-or-equal`

Эта функция должна принимать два аргумента типа данных "http://www.w3.org/2001/XMLSchema#time" "http://www.w3.org/2001/XMLSchema#boolean". Она должна возвращать "True", если и только если первый аргумент больше или равен второму аргументу в соответствии с отношением упорядочивания для "http://www.w3.org/2001/XMLSchema#time" (W3C Datatypes:2001, 3.2.8). Иначе, она должна возвращать "False".

ПРИМЕЧАНИЕ 2. – Считается, что нельзя сравнивать время, в которое включено значение временной зоны с тем временем, у которого такого значения нет. В таких случаях нужно использовать функцию время-в-диапазоне.

`urn:oasis:names:tc:xacml:1.0:function:time-less-than`

Эта функция должна принимать два аргумента типа данных "http://www.w3.org/2001/XMLSchema#time" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Она должна возвращать "True", если и только если первый аргумент меньше второго аргумента в соответствии с отношением упорядочивания, установленном для "http://www.w3.org/2001/XMLSchema#time" (W3C Datatypes:2001, 3.2.8). Иначе, она должна возвращать "False".

Otherwise, it shall return "False".

ПРИМЕЧАНИЕ 3. – Считается, что нельзя сравнивать время, в которое включено значение временной зоны с тем временем, у которого такого значения нет. В таких случаях нужно использовать функцию время-в-диапазоне.

`urn:oasis:names:tc:xacml:1.0:function:time-less-than-or-equal`

Эта функция должна принимать два аргумента типа данных "http://www.w3.org/2001/XMLSchema#time" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Она должна возвращать "True", если и только если первый аргумент меньше или равен второму аргументу в соответствии с отношением упорядочивания, установленном для "http://www.w3.org/2001/XMLSchema#time". Иначе, она должна возвращать "False".

ПРИМЕЧАНИЕ 4. – Считается, что нельзя сравнивать время, в которое включено значение временной зоны с тем временем, у которого такого значения нет. В таких случаях нужно использовать функцию время-в-диапазоне.

`urn:oasis:names:tc:xacml:1.0:function:time-in-range`

Эта функция должна принимать три аргумента типа данных "http://www.w3.org/2001/XMLSchema#time" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Она должна возвращать "True", если первый аргумент попадает в диапазон, определенный, включительно, вторым и третьим аргументами. Иначе, она должна возвращать "False". Независимо от своего значения, третий аргумент должен интерпретироваться, как время, которое равно или более позднее, чем менее чем двадцать четыре часа, второму аргументу. Если для первого аргумента не предоставлено никакой временной зоны, то он должен использовать временную зону по умолчанию в обработчике контекста. Если для второго и третьего аргумента не предоставлено никакой временной зоны, то они должны использовать временную зону из первого аргумента.

`urn:oasis:names:tc:xacml:1.0:function:dateTime-greater-than`

Эта функция должна принимать два аргумента типа данных "http://www.w3.org/2001/XMLSchema#dateTime" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Она должна возвращать "True", если и только если первый аргумент больше, чем второй аргумент в соответствии с отношением упорядочивания, установленном для "http://www.w3.org/2001/XMLSchema#dateTime" с помощью W3C Datatype:2001, 3.2.7. Иначе, она должна возвращать "False".

ПРИМЕЧАНИЕ 5. – Если в значение `dateTime` не включено значение временной зоны, то должно быть назначено подразумеваемое значение временной зоны, как описано в W3C Datatype:2001.

`urn:oasis:names:tc:xacml:1.0:function:dateTime-greater-than-or-equal`

Эта функция должна принимать два аргумента типа данных "http://www.w3.org/2001/XMLSchema#dateTime" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Она должна возвращать "True", если и только если первый аргумент больше или равен второму аргументу в соответствии с отношением упорядочивания, установленном для "http://www.w3.org/2001/XMLSchema#dateTime" с помощью W3C Datatype:2001, 3.2.7. Иначе, она должна возвращать "False".

ПРИМЕЧАНИЕ 6. – Если в значение `dateTime` не включено значение временной зоны, то должно быть назначено подразумеваемое значение временной зоны, как описано в W3C Datatype:2001.

`urn:oasis:names:tc:xacml:1.0:function:dateTime-less-than`

Эта функция должна принимать два аргумента типа данных "http://www.w3.org/2001/XMLSchema#dateTime" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Она должна возвращать "True", если и только если первый аргумент меньше, чем второй аргумент в соответствии с отношением упорядочивания, установленном для "http://www.w3.org/2001/XMLSchema#dateTime" с помощью W3C Datatype:2001, 3.2.7. Иначе, она должна возвращать "False".

ПРИМЕЧАНИЕ 7. – Если в значение dateTime не включено значение временной зоны, то должно быть назначено подразумеваемое значение временной зоны, как описано в W3C Datatype:2001.

```
urn:oasis:names:tc:xacml:1.0:function:dateTime-less-than-or-equal
```

Эта функция должна принимать два аргумента типа данных "http://www.w3.org/2001/XMLSchema# dateTime" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Она должна возвращать "True", если и только если первый аргумент меньше или равен второму аргументу в соответствии с отношением упорядочивания, установленном для "http://www.w3.org/2001/XMLSchema#dateTime" с помощью W3C Datatypes:2001, 3.2.7. Иначе она должна возвращать "False".

ПРИМЕЧАНИЕ 8. – Если в значение dateTime не включено значение временной зоны, то должно быть назначено подразумеваемое значение временной зоны, как описано в W3C Datatype:2001.

```
urn:oasis:names:tc:xacml:1.0:function:date-greater-than
```

Эта функция должна принимать два аргумента типа данных "http://www.w3.org/2001/XMLSchema#date" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Она должна возвращать "True", если и только если первый аргумент больше, чем второй аргумент в соответствии с отношением упорядочивания, установленном для "http://www.w3.org/2001/XMLSchema#date". Иначе она должна возвращать "False".

ПРИМЕЧАНИЕ 9. – Если в значение dateTime не включено значение временной зоны, то должно быть назначено подразумеваемое значение временной зоны, как описано в W3C Datatype:2001.

```
urn:oasis:names:tc:xacml:1.0:function:date-greater-than-or-equal
```

Эта функция должна принимать два аргумента типа данных "http://www.w3.org/2001/XMLSchema#date" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Она должна возвращать "True", если и только если первый аргумент больше или равен второму аргументу в соответствии с отношением упорядочивания, установленном для "http://www.w3.org/2001/XMLSchema#date". Иначе она должна возвращать "False".

ПРИМЕЧАНИЕ 10. – Если в значение dateTime не включено значение временной зоны, то должно быть назначено подразумеваемое значение временной зоны, как описано в W3C Datatype:2001.

```
urn:oasis:names:tc:xacml:1.0:function:date-less-than
```

Эта функция должна принимать два аргумента типа данных "http://www.w3.org/2001/XMLSchema#date" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Она должна возвращать "True", если и только если первый аргумент меньше, чем второй аргумент в соответствии с отношением упорядочивания, установленном для "http://www.w3.org/2001/XMLSchema#date". Иначе она должна возвращать "False".

ПРИМЕЧАНИЕ 11. – Если в значение dateTime не включено значение временной зоны, то должно быть назначено подразумеваемое значение временной зоны, как описано в W3C Datatype:2001.

```
urn:oasis:names:tc:xacml:1.0:function:date-less-than-or-equal
```

Эта функция должна принимать два аргумента типа данных "http://www.w3.org/2001/XMLSchema#date" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Она должна возвращать "True", если и только если первый аргумент меньше или равен второму аргументу в соответствии с отношением упорядочивания, установленном для "http://www.w3.org/2001/XMLSchema#date". Иначе она должна возвращать "False".

ПРИМЕЧАНИЕ 12. – Если в значение dateTime не включено значение временной зоны, то должно быть назначено подразумеваемое значение временной зоны, как описано в W3C Datatype:2001.

A.3.9 Функции строки

Следующие функции осуществляют операции со строками и идентификаторами URI.

```
urn:oasis:names:tc:xacml:2.0:function:string-concatenate
```

Эта функция должна принимать два аргумента типа данных "http://www.w3.org/2001/XMLSchema#string" и должна возвращать "http://www.w3.org/2001/XMLSchema#string". Результатом должно быть сцепление, по порядку, аргументов.

```
urn:oasis:names:tc:xacml:2.0:function:url-string-concatenate
```

Эта функция должна принимать один аргумент типа данных "http://www.w3.org/2001/XMLSchema#anyURI" и один или более аргументов типа "http://www.w3.org/2001/XMLSchema#string", и должна возвращать "http://www.w3.org/2001/XMLSchema#anyURI". Результатом должен быть URI, построенный прикреплением, по порядку, аргументов "string" к аргументу "anyURI".

A.3.10 Функции "мешка"

Эти функции осуществляют операции с "мешками" значений "type", где тип является одним из простых типов данных. Некоторые дополнительные условия, определенные для каждой функции ниже, должны послужить причиной оценки выражения, как "Indeterminate".

`urn:oasis:names:tc:xacml:1.0:function:type-one-and-only`

Эта функция должна принимать "мешок" значений "type" в качестве аргумента и должна возвращать значение "-type". Она должна возвращать только одно значение в "мешок". Если в "мешке" нет одного и только одного значения, то это выражение должно оцениваться, как "Indeterminate".

`urn:oasis:names:tc:xacml:1.0:function:type-bag-size`

Эта функция должна принимать "мешок" значений "type" в качестве аргумента и должна возвращать "http://www.w3.org/2001/XMLSchema#integer", указывая количество значений в "мешке".

`urn:oasis:names:tc:xacml:1.0:function:type-is-in`

Эта функция должна принимать аргумент "type" в качестве своего первого аргумента и "мешок" значений типа в качестве своего второго аргумента, и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Функция должна оценивать в "True", если и только если первый аргумент сопоставим с помощью "`urn:oasis:names:tc:xacml:x.x:function:type-equal`" с любым значением в этом "мешке". Иначе она должна возвращать "False".

`urn:oasis:names:tc:xacml:1.0:function:type-bag`

Эта функция должна принимать любое количество аргументов "type" и возвращать "мешок" значений "type", содержащий значения аргументов. Применение этой функции к нулю аргументов должно привести к появлению пустого "мешка", установленного типа данных.

A.3.11 Функции набора

Эти функции осуществляют операции с "мешками" мнемонических наборов с помощью исключения повторяющихся элементов из "мешка".

`urn:oasis:names:tc:xacml:1.0:function:type-intersection`

Эта функция должна принимать два аргумента, и они оба являются "мешком" значений "type". Она должна возвращать "мешок" значений "type", таким образом, чтобы в нем содержались только элементы, которые являются общими для этих двух "мешков", что определяется с помощью "`urn:oasis:names:tc:xacml:x.x:function:type-equal`". В результате не должно быть повторяющихся значений, как определено с помощью "`urn:oasis:names:tc:xacml:x.x:function:type-equal`".

`urn:oasis:names:tc:xacml:1.0:function:type-at-least-one-member-of`

Эта функция должна принимать два аргумента, и они оба являются "мешком" значений "type". Она должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Эта функция должна оцениваться, как "True", если и только если, по крайней мере, один элемент первого аргумента содержится во втором аргументе, как определено с помощью "`urn:oasis:names:tc:xacml:x.x:function:type-is-in`".

`urn:oasis:names:tc:xacml:1.0:function:type-union`

Эта функция должна принимать два аргумента, и они оба являются "мешком" значений "type". Это выражение должно возвращать "мешок" "type", таким образом, чтобы в нем содержались все элементы обоих "мешков". В результате не должно быть повторяющихся значений, как определено с помощью "`urn:oasis:names:tc:xacml:x.x:function:type-equal`".

`urn:oasis:names:tc:xacml:1.0:function:type-subset`

Эта функция должна принимать два аргумента, и они оба являются "мешком" значений "type". Она должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Она должна возвращать "True", если и только если первый аргумент является подмножеством второго аргумента. Должно считаться, что повторяющиеся элементы удалены из каждого аргумента до вычисления подмножества, как определено с помощью "`urn:oasis:names:tc:xacml:x.x:function:type-equal`".

`urn:oasis:names:tc:xacml:1.0:function:type-set-equals`

Эта функция должна принимать два аргумента, и они оба являются "мешком" значений "type". Она должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Она должна возвращать результат, применяемого "`urn:oasis:names:tc:xacml:1.0:function:and`" приложению "`urn:oasis:names:tc:xacml:x.x:function:type-subset`" к первому и второму аргументам и приложению "`urn:oasis:names:tc:xacml:x.x:function:type-subset`" ко втором и первому аргументам.

A.3.12 Функции "мешка" высшего порядка

В этом пункте обсуждаются функции в XACML, которые выполняют операции над "мешками" таким образом, чтобы эти функции можно было применить к "мешкам" в общем.

Функциональный язык общего назначения был бы полезен для установления семантики этих функций (смотрите Приложение III для рассмотрения информативного примера использования функционального языка).

1) `urn:oasis:names:tc:xacml:1.0:function:any-of`

   Эта функция применяет логическую функцию между конкретным значением примитива и "мешком" значений, и должна возвращать "True", если и только если предикат имеет значение "True" для, по крайней мере, одного элемента в "мешке".

   Эта функция должна принимать три аргумента. Первым аргументом должен быть элемент <xacml:Function>, который дает имя логической функции, принимающей два аргумента простых типов. Вторым аргументом должно быть значение простого типа данных. Третьим аргументом должен быть "мешок" простого типа данных. Выражение должно оцениваться таким образом, как если бы функция, именованная в аргументе <xacml:Function>, была бы применена ко второму аргументу и каждому элементу третьего аргумента ("мешок"), и результаты были бы объединены с помощью "urn:oasis:names:tc:xacml:1.0:function:or".

2) `urn:oasis:names:tc:xacml:1.0:function:all-of`

   Эта функция применяет логическую функцию между конкретным значением примитива и "мешком" значений, и должна возвращать "True", если и только если предикат имеет значение "True" для каждого элемента в "мешке".

   Эта функция должна принимать три аргумента. Первым аргументом должен быть элемент <xacml:Function>, который дает имя логической функции, принимающей два аргумента простых типов. Вторым аргументом должно быть значение простого типа данных. Третьим аргументом должен быть "мешок" простого типа данных. Выражение должно оцениваться таким образом, как если бы функция, именованная в аргументе <xacml:Function>, была бы применена ко второму аргументу и каждому элементу третьего аргумента ("мешок"), и результаты были бы объединены с помощью "urn:oasis:names:tc:xacml:1.0:function:and".

3) `urn:oasis:names:tc:xacml:1.0:function:any-of-any`

   Эта функция применяет логическую функцию между каждым элементом "мешка" значений и каждым элементом другого "мешка" значений, и возвращает "True", если и только если предикат имеет значение "True" для, по крайней мере, одного сравнения.

   Эта функция должна принимать три аргумента. Первым аргументом должен быть элемент <xacml:Function>, который дает имя логической функции, принимающей два аргумента простых типов. Вторым аргументом должно быть значение простого типа данных. Третьим аргументом должен быть "мешок" простого типа данных. Выражение должно оцениваться таким образом, как если бы функция, именованная в аргументе <xacml:Function>, была бы применена к каждому элементу второго аргумента и каждому элементу третьего аргумента, и результаты были бы объединены с помощью "urn:oasis:names:tc:xacml:1.0:function:or". Семантика такова: результат выражения должен быть "True", если и только если примененный предикат имеет значение "True" для, по крайней мере, одного сравнения элементов из двух "мешков".

4) urn:oasis:names:tc:xacml:1.0:function:all-of-any

   Эта функция применяет логическую функцию между элементами из двух "мешков". Выражение должно быть "True", если и только если предоставленный предикат имеет значение 'True' между каждым элементом первого "мешка" и любым элементом второго "мешка".

   Эта функция должна принимать три аргумента. Первым аргументом должен быть элемент <xacml:Function>, который дает имя логической функции, принимающей два аргумента простых типов. Вторым аргументом должен быть "мешок" простого типа данных. Третьим аргументом должен быть "мешок" простого типа данных. Выражение должно оцениваться таким образом, как если бы функция "urn:oasis:names:tc:xacml:1.0:function:any-of" была бы применена к каждому значению первого "мешка" и ко всему второму "мешку", используя предоставленный xacml:Function, и результаты были бы затем объединены с использованием "urn:oasis:names:tc:xacml:1.0:function:and".

5) `urn:oasis:names:tc:xacml:1.0:function:any-of-all`

   Эта функция применяет логическую функцию между элементами из двух "мешков". Выражение должно быть "True", если и только если предоставленный предикат имеет значение 'True' между каждым элементом второго "мешка" и любым элементом первого "мешка".

   Эта функция должна принимать три аргумента. Первым аргументом должен быть элемент <xacml:Function>, который дает имя логической функции, принимающей два аргумента простых типов. Вторым аргументом должен быть "мешок" простого типа данных. Третьим аргументом должен быть "мешок" простого типа данных. Выражение должно оцениваться таким

образом, как если бы функция "urn:oasis:names:tc:xacml:1.0:function:any-of" была бы применена к каждому значению второго "мешка" и ко всему первому "мешку", используя предоставленный xacml:Function, и результаты были бы затем объединены с использованием "urn:oasis:names:tc:xacml:1.0:function:and".

6) urn:oasis:names:tc:xacml:1.0:function:all-of-all

Эта функция применяет логическую функцию между элементами из двух "мешков". Выражение должно быть "True", если и только если предоставленный предикат имеет значение 'True' между всеми вместе элементами первого "мешка" и всеми элементами второго "мешка".

Эта функция должна принимать три аргумента. Первым аргументом должен быть элемент <xacml:Function>, который дает имя логической функции, принимающей два аргумента простых типов. Вторым аргументом должен быть "мешок" простого типа данных. Третьим аргументом должен быть "мешок" простого типа данных. Выражение должно оцениваться таким образом, как если бы функция, именованная в элементе <xacml:Function>, была приложена между каждым элементом второго аргумента и каждым элементом третьего аргумента, и результаты были бы объединены с помощью "urn:oasis:names:tc:xacml:1.0:function:and". Семантика такова: результат выражения должен быть "True", если и только если примененный предикат имеет значение "True" для всех элементов первого "мешка", сравниваемого со всеми элементами второго "мешка".

7) urn:oasis:names:tc:xacml:1.0:function:map

Эта функция преобразует "мешок" значений в другой "мешок" значений.

Эта функция должна принимать два аргумента. Первой функцией должен быть элемент <xacml:Function>, именующий функцию, которая принимает единый аргумент простого типа данных и возвращает значение простого типа данных. Вторым аргументом должен быть "мешок" простого типа данных. Выражение должно оцениваться таким образом, как если бы функция, именованная в элементе <xacml:Function>, была приложена к каждому элементу в "мешке" и в результате образовывался бы "мешок" преобразованного значения. Результатом должен быть "мешок" простого типа данных, который возвращается с помощью функции, именованной в элементе <xacml:Function>.

A.3.13 Функции, основанные на регулярных выражениях

Эти функции осуществляют операции над различными типами, используя регулярные выражения, и оцениваются как "http://www.w3.org/2001/XMLSchema#boolean".

urn:oasis:names:tc:xacml:1.0:function:string-regexp-match

Эта функция решает сопоставление регулярного выражения. Она должна принимать два аргумента "http://www.w3.org/2001/XMLSchema#string" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Первым аргументом должно быть регулярное выражение, вторым аргументом должна быть главная строка. Функция должна возвращать "True", если и только если второй аргумент сопоставим со значением образца регулярного выражения в первом аргументе. Синтаксис регулярного выражения должен быть таким, как определено в W3C Datatypes:2001, Приложение F, расширенный с помощью следующих дополнительных выражений и правил образца:

- X?? сопоставим с X не более одного раза
- X*? сопоставим с X любое количество раз, включая ноль
- X+? сопоставим с X, по крайней мере, один раз
- X{n}? сопоставим с X ровно n раз
- X(n,}? сопоставим с X, по крайней мере, n раз
- X{n,m}? сопоставим с X, по крайней мере, n раз, но не более m раз

Если используется одно из этих дополнительных выражений образцов, то регулярное выражение должно быть сопоставимо с самой короткой возможной подстрокой первого аргумента, которая согласована с этим образцом.

За исключением этих дополнительных выражений, регулярное выражение должно быть сопоставимо с самой длинной возможной подстрокой первого аргумента, которая согласована с этим образцом.

За исключением случаев, когда образец эксплицитно закреплен в начале или в конце строки, с использованием символов образца "^" и "$", соответственно, этот образец считается сопоставимым, если он сопоставим с любой подстрокой второго аргумента.

Все совместимые реализации должны поддерживать установленные образцы регулярных выражений. Совместимые реализации могут поддерживать дополнительные образцы регулярных выражений.

urn:oasis:names:tc:xacml:2.0:function:anyURI-regexp-match

Эта функция решает сопоставление регулярного выражения. Она должна принимать два аргумента; первым является аргумент типа "http://www.w3.org/2001/XMLSchema#string" и вторым является аргумент типа "http://www.w3.org/2001/XMLSchema#anyURI". Она должна возвращать "http://www.w3.org/2001/XMLSchema#boolean".

Первый аргумент должен быть регулярным выражением, а вторым аргументом должен быть URI. Эта функция должна преобразовать второй аргумент в тип "http://www.w3.org/2001/XMLSchema#string", а затем применить "urn:oasis:names:tc:xacml:1.0:function:string-regexp-match".

urn:oasis:names:tc:xacml:2.0:function:ipAddress-regexp-match

Эта функция решает сопоставление регулярного выражения. Она должна принимать два аргумента; первым является аргумент типа "http://www.w3.org/2001/XMLSchema#string" и вторым является аргумент "urn:oasis:names:tc:xacml:2.0:data-type:ipAddress". Она должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Первый аргумент должен быть регулярным выражением, а вторым аргументом должен быть адрес IPv4 или IPv6. Эта функция должна преобразовать второй аргумент в тип "http://www.w3.org/2001/XMLSchema#string", а затем применить "urn:oasis:names:tc:xacml:1.0:function:string-regexp-match".

urn:oasis:names:tc:xacml:2.0:function:dnsName-regexp-match

Эта функция решает сопоставление регулярного выражения. Она должна принимать два аргумента; первым является аргумент типа "http://www.w3.org/2001/XMLSchema#string" и вторым является аргумент "urn:oasis:names:tc:xacml:2.0:data-type:dnsName". Она должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Первый аргумент должен быть регулярным выражением, а вторым аргументом должно быть имя DNS. Эта функция должна преобразовать второй аргумент в тип "http://www.w3.org/2001/XMLSchema#string", а затем применить "urn:oasis:names:tc:xacml:1.0:function:string-regexp-match".

urn:oasis:names:tc:xacml:2.0:function:rfc822Name-regexp-match

Эта функция решает сопоставление регулярного выражения. Она должна принимать два аргумента; первым является аргумент типа "http://www.w3.org/2001/XMLSchema#string" и вторым является аргумент "urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name". Она должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Первый аргумент должен быть регулярным выражением, а вторым аргументом должен быть RFC имя 822. Эта функция должна преобразовать второй аргумент в тип "http://www.w3.org/2001/XMLSchema#string", а затем применить "urn:oasis:names:tc:xacml:1.0:function:string-regexp-match".

urn:oasis:names:tc:xacml:2.0:function:x500Name-regexp-match

Эта функция решает сопоставление регулярного выражения. Она должна принимать два аргумента; первым является аргумент типа "http://www.w3.org/2001/XMLSchema#string" и вторым является аргумент "urn:oasis:names:tc:xacml:1.0:data-type:x500Name". Она должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Первый аргумент должен быть регулярным выражением, а вторым аргументом должен быть именем справочника X.500. Эта функция должна преобразовать второй аргумент в тип "http://www.w3.org/2001/XMLSchema#string", а затем применить "urn:oasis:names:tc:xacml:1.0:function:string-regexp-match".

A.3.14 Особые функции сопоставления

Эти функции осуществляют операции над различными типами и оцениваются как "http://www.w3.org/2001/XMLSchema#boolean", основываясь на установленных стандартных алгоритмах сопоставления.

urn:oasis:names:tc:xacml:1.0:function:x500Name-match

Эта функция должна принимать два аргумента "urn:oasis:names:tc:xacml:2.0:data-type:x500Name" и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Она должна возвращать "True", если и только если первый аргумент сопоставим с какой-либо заключительной последовательностью имен RDN из второго аргумента, если сравнение происходит с использованием x500Name-equal.

urn:oasis:names:tc:xacml:1.0:function:rfc822Name-match

Эта функция должна принимать два аргумента; первым является тип данных "http://www.w3.org/2001/XMLSchema#string" и вторым является тип данных "urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name", и должна возвращать "http://www.w3.org/2001/XMLSchema#boolean". Эта функция должна оцениваться в значение "True", если первый аргумент сопоставим со вторым аргументом, в соответствии с W3C Datatypes:2001, 3.2.1.

IETF RFC 822 name состоит из местной части, за которой следует "@", а далее следует часть домена. Местная часть зависит от конкретных условий, в то время как часть домена (которая обычно является именем главного компьютера DNS) не является зависимой от конкретных условий.

Второй аргумент содержит полное rfc822Name. Первым аргументом является полное или частичное rfc822Name, используемое для выбора подходящих значений во втором аргументе следующим образом.

Для того чтобы быть сопоставимым с конкретным адресом во втором аргументе, первый аргумент должен установить полный почтовый адрес, с которым будет проводиться сравнение. Для того чтобы сопоставить любой адрес в конкретном домене со вторым аргументом, первый аргумент должен установить только имя домена (обычно имя DNS). Для того чтобы сопоставить любой адрес в конкретном домене во втором аргументе, первый аргумент должен установить желаемую часть домена с начальным символом ".".

A.3.15 Функции, основанные на XPath

В этом пункте устанавливаются функции, которые принимают выражения XPath в качестве аргументов. Выражение XPath оценивается как набор узлов, который является набором узлов XML, сопоставимым с этим выражением. Узел или набор узлов не является формальной системой типов данных XACML. Все сравнения или другие операции над набором узлов выполняются независимо от установленной конкретной функции. Другими словами, выражения XPath в этих функциях ограничены контекстом запроса XACML. Элемент `<xacml-context:Request>` является узлом контекста для каждого выражения XPath. Определены следующие функции:

```
urn:oasis:names:tc:xacml:1.0:function:xpath-node-count
```

Эта функция должна принимать "http://www.w3.org/2001/XMLSchema#string", в качестве аргумента, который дожжен интерпретироваться, как выражение XPath, и оцениваться, как "http://www.w3.org/2001/XMLSchema#integer". Значение, возвращаемое из функции, должно быть подсчетом узлов внутри набора узлов, которые сопоставимы с заданным выражением XPath.

```
urn:oasis:names:tc:xacml:1.0:function:xpath-node-equal
```

Эта функция должна принимать два аргумента "http://www.w3.org/2001/XMLSchema#string", которые должны интерпретироваться, как выражения XPath, и должны возвращать "http://www.w3.org/2001/XMLSchema#boolean". Эта функция должна возвращать "True", если только любой узел XML в наборе узлов, сопоставленный с помощью первого аргумента, равен любому узлу XML в наборе узлов, сопоставленному с помощью второго аргумента. Два узла считаются равными, если у них одна и та же идентичность.

```
urn:oasis:names:tc:xacml:1.0:function:xpath-node-match
```

Эта функция должна принимать два аргумента "http://www.w3.org/2001/XMLSchema#string", которые должны интерпретироваться, как выражения XPath, и должны возвращать "http://www.w3.org/2001/XMLSchema#boolean". Эта функция должна оцениваться, как "True", если удовлетворено одно из двух следующих условий:

1) Любой узел XML в наборе узлов, сопоставленном с помощью первого аргумента, равен любому узлу XML в наборе узлов, сопоставленному с помощью второго аргумента;

2) любой атрибут и узел элемента ниже любого узла XML в наборе узлов, сопоставленном с помощью первого аргумента, равен любому узлу XML в наборе узлов, сопоставленному с помощью второго аргумента.

Два узла считаются равными, если у них одна и та же идентичность.

ПРИМЕЧАНИЕ. – Первое условие эквивалентно "xpath-node-equal", и гарантирует, что "xpath-node-equal" – это частный случай "xpath-node-match".

A.3.16 Функции расширения и простые типы

Функции и простые типы устанавливаются с помощью строковых идентификаторов, разрешенных для представления функций, дополнительно к тем, которые установлены языком XACML. Такой подход разрешает любому расширить модуль XACML особыми функциями и особыми простыми типами данных.

Для того чтобы сохранить целостность стратегии оценки XACML, результат функции расширения должен зависеть только от значений ее аргументов. Глобальные и скрытые параметры не должны влиять на оценку выражения. У функций не должно быть побочных эффектов, иначе нельзя гарантировать стандартный порядок оценки.

Приложение B

Идентификаторы XACML

В этом приложении определяются стандартные идентификаторы для часто используемых объектов.

B.1 Пространство имен XACML

В настоящее время существует два определенных пространства имен XACML.

Стратегии определяются с использованием этого идентификатора.

```
urn:oasis:names:tc:xacml:2.0:policy:schema:os
```

Контексты запросов и ответов определяются с использованием этого идентификатора.

```
urn:oasis:names:tc:xacml:2.0:context:schema:os
```

B.2 Категории субъектов доступа

Этот идентификатор показывает системный объект, который инициирует запрос о доступе. Другими словами, инициирующий объект в цепи запроса. Если категория субъекта не установлена, то это значение является значением по умолчанию.

```
urn:oasis:names:tc:xacml:1.0:subject-category:access-subject
```

Этот идентификатор показывает системный объект, который получит результаты запроса (используется, если он отличается от субъекта доступа).

```
urn:oasis:names:tc:xacml:1.0:subject-category:recipient-subject
```

Этот идентификатор показывает системный объект, через который прошел запрос о доступе. Их может быть больше одного. Не предоставлено способов определения порядка прохождения сообщения через них.

```
urn:oasis:names:tc:xacml:1.0:subject-category:intermediary-subject
```

Этот идентификатор показывает системный объект, связанный с местным или удаленным основанием кода, который создал этот запрос. В соответствующие атрибуты субъекта может быть включен URL, от которого он был получен и/или идентичность подписанта кода. Их может быть больше одного. Не предоставлено способов определения порядка обработки сообщения.

```
urn:oasis:names:tc:xacml:1.0:subject-category:codebase
```

Этот идентификатор показывает системный объект, связанный с компьютером, который инициировал запрос ***access***. Примером могла бы быть идентичность IPSEC.

```
urn:oasis:names:tc:xacml:1.0:subject-category:requesting-machine
```

B.3 Типы данных

Следующие идентификаторы показывают типы данных, которые определены в A.2.

```
urn:oasis:names:tc:xacml:2.0:data-types:dayTimeDuration
urn:oasis:names:tc:xacml:2.0:data-types:yearMonthDuration
urn:oasis:names:tc:xacml:1.0:data-type:x500Name.
urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name
urn:oasis:names:tc:xacml:2.0:data-type:ipAddress
urn:oasis:names:tc:xacml:2.0:data-type:dnsName
```

Следующие идентификаторы типов данных определены с помощью W3C DataTypes:2001.

```
http://www.w3.org/2001/XMLSchema#string
http://www.w3.org/2001/XMLSchema#boolean
http://www.w3.org/2001/XMLSchema#integer
http://www.w3.org/2001/XMLSchema#double
http://www.w3.org/2001/XMLSchema#time
http://www.w3.org/2001/XMLSchema#date
http://www.w3.org/2001/XMLSchema#dateTime
```

```
http://www.w3.org/2001/XMLSchema#anyURI
http://www.w3.org/2001/XMLSchema#hexBinary
http://www.w3.org/2001/XMLSchema#base64Binary
```

B.4 Атрибуты субъекта

Эти идентификаторы показывают атрибуты субъекта. Если они используются, то они должны появляться внутри элемента <Subject> контекста запроса. К ним должен быть предоставлен доступ с помощью средств элемента <SubjectAttributeDesignator> или элемента <AttributeSelector>, который указывает внутрь элемента <Subject> контекста запроса.

Самое большее, один из этих атрибутов связан с каждым субъектом. Каждый атрибут, связанный с аутентификацией, включенный в единый элемент <Subject>, связан с тем же самым событием аутентификации.

Этот идентификатор показывает имя субъекта. Форматом по умолчанию является "http://www.w3.org/2001/XMLSchema#string". Для указания на другие форматы используйте атрибуты DataType, перечисленные в B.3.

```
urn:oasis:names:tc:xacml:1.0:subject:subject-id
```

Этот идентификатор показывает категорию субъекта. Значением по умолчанию является "access-subject".

```
urn:oasis:names:tc:xacml:1.0:subject-category
```

Этот идентификатор показывает домен безопасности субъекта. Он идентифицирует администратора и стратегию, которая управляет пространством имен, в котором происходит управление id субъекта.

```
urn:oasis:names:tc:xacml:1.0:subject:subject-id-qualifier
```

Этот идентификатор показывает открытый ключ, использующийся для подтверждения идентичности субъекта.

```
urn:oasis:names:tc:xacml:1.0:subject:key-info
```

Этот идентификатор показывает время, когда субъект был аутентифицирован.

```
urn:oasis:names:tc:xacml:1.0:subject:authentication-time
```

Этот идентификатор показывает метод, используемый для аутентификации субъекта.

```
urn:oasis:names:tc:xacml:1.0:subject:authn-locality:authentication-method
```

Этот идентификатор показывает время, когда субъект инициировал запрос о доступе, в соответствии с PEP.

```
urn:oasis:names:tc:xacml:1.0:subject:request-time
```

Этот идентификатор показывает время, когда начался текущий сеанс субъекта, в соответствии с PEP.

```
urn:oasis:names:tc:xacml:1.0:subject:session-start-time
```

Следующий идентификатор показывает местоположение, где были активированы мандаты аутентификации. Они предназначены для поддержания соответствующих объектов из заявления об аутентификации SAML.

Этот идентификатор показывает, что местоположение выражено, как адрес IP.

```
urn:oasis:names:tc:xacml:1.0:subject:authn-locality:ip-address
```

Соответствующий атрибут должен быть такого типа данных "http://www.w3.org/2001/XMLSchema#string".

Этот идентификатор показывает, что местоположение выражено в имени DNS.

```
urn:oasis:names:tc:xacml:1.0:subject:authn-locality:dns-name
```

Соответствующий атрибут должен быть такого типа данных "http://www.w3.org/2001/XMLSchema#string".

Если подходящий атрибут уже определен в LDAP, то идентификатор XACML должен быть сформирован добавлением имени атрибута к URI IETF LDAP RFC (смотрите IETF RFC 2256). Например, имя атрибута для userPassword, определенное в IETF RFC 2256, должно быть:

```
http://www.ietf.org/rfc/rfc2256.txt#userPassword
```

B.5 Атрибуты ресурса

Эти идентификаторы показывают атрибуты ресурса. Соответствующие атрибуты могут появляться в элементе <Resource> контекста запроса и доступ к ним может быть получен с помощью элемента <ResourceAttributeDesignator> или с помощью элемента <AttributeSelector>, который указывает внутрь элемента <Resource> контекста запроса.

Этот атрибут показывает ресурс, к которому запрашивается доступ. Если предоставлен элемент <xacml-context:ResourceContent>, то ресурсом, к которому запрашивается доступ, должен быть полностью или частично тот ресурс, который предоставлен в элементе <xacml-context:ResourceContent>.

```
urn:oasis:names:tc:xacml:1.0:resource:resource-id
```

Этот атрибут идентифицирует пространство имен высшего элемента содержания элемента <xacml-context:ResourceContent>. В случае, если содержание ресурса предоставлено в контексте запроса и пространство имен ресурса определено в этом ресурсе, то PDP должен подтвердить, что пространство имен, определенное в этом атрибуте то же самое, что и то, которое определено в этом ресурсе. Типом соответствующего атрибута должен быть "http://www.w3.org/2001/XMLSchema#anyURI".

```
urn:oasis:names:tc:xacml:2.0:resource:target-namespace
```

B.6 Атрибуты действия

Эти идентификаторы показывают атрибуты запрашиваемого действия. Если они используются, то они должны появиться внутри элемента <Action> контекста запроса. Доступ к ним должен быть предоставлен с помощью элемента <ActionAttributeDesignator> или элемента <AttributeSelector>, который указывает внутрь элемента <Action> контекста запроса.

Этот атрибут идентифицирует действие, для которого запрашивается доступ.

```
urn:oasis:names:tc:xacml:1.0:action:action-id
```

Если действие неявное, то значением атрибута action-id должно быть:

```
urn:oasis:names:tc:xacml:1.0:action:implied-action
```

Этот атрибут идентифицирует пространство имен, в котором определен атрибут action-id.

```
urn:oasis:names:tc:xacml:1.0:action:action-namespace
```

B.7 Атрибуты среды

Эти идентификаторы показывают атрибуты среды, внутри которой должен оцениваться запрос о принятии решения. Если они используются в запросе о принятии решения, то они должны появиться в элементе <Environment> контекста запроса. Доступ к ним должен быть предоставлен с помощью элемента <EnvironmentAttributeDesignator> или элемента <AttributeSelector>, который указывает внутрь элемента <Environment> контекста запроса.

Этот идентификатор показывает текущее время в обработчике контекста. Практически, это то время, когда был создан контекст запроса. По этой причине, если эти идентификаторы появляются во многих местах внутри <Policy> или <PolicySet>, то это же самое значение должно быть назначено каждому эпизоду в процедуре оценки, независимо от того, сколько времени прошло между обработкой этих эпизодов.

```
urn:oasis:names:tc:xacml:1.0:environment:current-time
```

Соответствующий атрибут должен быть такого типа данных "http://www.w3.org/2001/XMLSchema#time".

```
urn:oasis:names:tc:xacml:1.0:environment:current-date
```

Соответствующий атрибут должен быть такого типа данных "http://www.w3.org/2001/XMLSchema#date".

```
urn:oasis:names:tc:xacml:1.0:environment:current-dateTime
```

Соответствующий атрибут должен быть такого типа данных "http://www.w3.org/2001/XMLSchema#dateTime".

B.8 Коды состояния

Определены следующие значения кода состояния.

Этот идентификатор показывает успех.

```
urn:oasis:names:tc:xacml:1.0:status:ok
```

Этот идентификатор показывает, что все атрибуты, необходимые для создания решения стратегии, были недоступны.

```
urn:oasis:names:tc:xacml:1.0:status:missing-attribute
```

Этот идентификатор показывает, что в каком-то значении атрибута, содержится ошибка синтаксиса, такая как буква в числовом поле.

```
urn:oasis:names:tc:xacml:1.0:status:syntax-error
```

Этот идентификатор показывает, что во время оценки стратегии произошла ошибка. Примером может быть деление на ноль.

```
urn:oasis:names:tc:xacml:1.0:status:processing-error
```

B.9 Алгоритмы объединения

У алгоритма объединения правил deny-overrides (запрет замен) имеется следующее значение для атрибута `ruleCombiningAlgId`:

```
urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides
```

У алгоритма объединения стратегий deny-overrides (запрет замен) имеется следующее значение для атрибута `policyCombiningAlgId`:

```
urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides
```

У алгоритма объединения правил permit-overrides (разрешение замен) имеется следующее значение для атрибута `ruleCombiningAlgId`:

```
urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides
```

У алгоритма объединения стратегий permit-overrides (разрешение замен) имеется следующее значение для атрибута `policyCombiningAlgId`:

```
urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:permit-overrides
```

У алгоритма объединения правил first-applicable (первый применим) имеется следующее значение для атрибута `ruleCombiningAlgId`:

```
urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-applicable
```

У алгоритма объединения стратегий first-applicable (первый применим) имеется следующее значение для атрибута `policyCombiningAlgId`:

```
urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:first-applicable
```

У алгоритма объединения стратегий only-one-applicable (только один применим) имеется следующее значение для атрибута `policyCombiningAlgId`:

```
urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:only-one-applicable
```

У алгоритма объединения правил ordered-deny-overrides (упорядоченный запрет замен) имеется следующее значение для атрибута `ruleCombiningAlgId`:

```
urn:oasis:names:tc:xacml:1.1:rule-combining-algorithm:ordered-deny-overrides
```

У алгоритма объединения стратегий ordered-deny-overrides (упорядоченный запрет замен) имеется следующее значение для атрибута `policyCombiningAlgId`:

```
urn:oasis:names:tc:xacml:1.1:policy-combining-algorithm:ordered-deny-overrides
```

У алгоритма объединения правил ordered-permit-overrides (упорядоченное разрешение замен) имеется следующее значение для атрибута `ruleCombiningAlgId`:

```
urn:oasis:names:tc:xacml:1.1:rule-combining-algorithm:ordered-permit-overrides
```

У алгоритма объединения стратегий ordered-permit-overrides (упорядоченное разрешение замен) имеется следующее значение для атрибута `policyCombiningAlgId`:

```
urn:oasis:names:tc:xacml:1.1:policy-combining-algorithm:ordered-permit-
overrides
```

Приложение C

Алгоритмы объединения

В этом приложении содержится описание алгоритмов объединения правил и стратегий, установленных языком XACML.

C.1 Deny-overrides (Запрет замен)

C.1.1 В этом пункте определяется алгоритм объединения правил "Deny-overrides" стратегии.

Во всем наборе правил в стратегии, если какое-либо правило оценивается как "Deny", то результатом объединения правила должно быть "Deny". Если любое из правил оценивается как "Permit", а все другие правила оцениваются как "NotApplicable", то результатом объединения правила должно быть "Permit". Другими словами, у "Deny" есть превосходство, независимо от результата оценки любого другого правила в этом объединении. Если обнаруживается, что все правила "NotApplicable" к запросу о принятии решения, то объединение правил должно оцениваться, как "NotApplicable".

Если произошла ошибка в процессе оценки цели или условия правила, в которой содержится значение эффекта "Deny", то процесс оценки должен продолжиться, чтобы оценить последующие правила, в поиске результата "Deny". Если больше ни одно правило не оценивается как "Deny", то объединение должно оцениваться, как "Indeterminate", с соответствующим статусом ошибки.

Если, по крайней мере, одно правило оценивается как "Permit", то все другие правила, у которых нет ошибок оценки, оцениваются как "Permit" или "NotApplicable", и все правила, в которых нет ошибок оценки, содержащих эффекты "Permit", то результатом объединения правил должно быть "Permit".

Следующий псевдокод представляет стратегию оценки этого алгоритма объединения правил.

```
Decision denyOverridesRuleCombiningAlgorithm(Rule rule[])
{
   Boolean atLeastOneError = false;
   Boolean potentialDeny  = false;
   Boolean atLeastOnePermit = false;
   for( i=0 ; i < lengthOf(rules) ; i++ )
   {
          Decision decision = evaluate(rule[i]);
          if (decision == Deny)
          {
                 return Deny;
          }
          if (decision == Permit)
          {
                 atLeastOnePermit = true;
                 continue;
          }
          if (decision == NotApplicable)
          {
                 continue;
          }
          if (decision == Indeterminate)
          {
                 atLeastOneError = true;

                 if (effect(rule[i]) == Deny)
                 {
                        potentialDeny = true;
                 }
                 continue;
          }
   }
   if (potentialDeny)
   {
          return Indeterminate;
   }
   if (atLeastOnePermit)
   {
          return Permit;
   }
   if (atLeastOneError)
```

```
    {
          return Indeterminate;
    }
    return NotApplicable;
}
```

C.1.2 В этом пункте определяется алгоритм объединения стратегий "Deny-overrides" стратегического набора.

Во всем наборе стратегий в стратегическом наборе, если какая-либо стратегия оценивается как "Deny", то результатом объединения стратегии должно быть "Deny". Другими словами, у "Deny" есть преимущество, независимо от результата оценки любой другой стратегии в стратегическом наборе. Если обнаруживается, что все стратегии "NotApplicable" к запросу о принятии решения, то этот стратегический набор должен оцениваться, как "NotApplicable".

Если произошла ошибка в процессе оценки цели стратегии или ссылка на стратегию считается неверной, или результатом оценки стратегии является "Indeterminate", то стратегический набор должен оцениваться, как"Deny".

Следующий псевдокод представляет стратегию оценки этого алгоритма объединения стратегии.

```
Decision denyOverridesPolicyCombiningAlgorithm(Policy policy[])
{
    Boolean atLeastOnePermit = false;
    for( i=0 ; i < lengthOf(policy) ; i++ )
    {
          Decision decision = evaluate(policy[i]);
          if (decision == Deny)
          {
                return Deny;
          }
          if (decision == Permit)
          {
                atLeastOnePermit = true;
                continue;
          }
          if (decision == NotApplicable)
          {
                continue;
          }
          if (decision == Indeterminate)
          {
                return Deny;
          }
    }
    if (atLeastOnePermit)
    {
          return Permit;
    }
    return NotApplicable;
}
```

Обязательства отдельных стратегий должны объединяться, как описано в пункте 7.6.14.

C.2 Ordered-deny-overrides (Упорядоченный запрет замен)

В следующем параграфе определяется алгоритм объединения правил "Ordered-deny-overrides" стратегии.

Поведение этого алгоритма идентично алгоритму объединения правил Deny-overrides с одним исключением. Порядок проведения оценки совокупности правил должен быть сопоставим с порядком, перечисленным в стратегии.

В следующем параграфе описывается алгоритм объединения стратегий "Ordered-deny-overrides" стратегического набора.

Поведение этого алгоритма идентично алгоритму объединения стратегий Deny-overrides с одним исключением. Порядок проведения оценки совокупности стратегий должен быть сопоставим с порядком, перечисленным в стратегическом наборе.

C.3 Permit-overrides (Разрешение замен)

C.3.1 В этом пункте определяется алгоритм объединения правил "Deny-overrides" стратегии.

Во всем наборе правил в стратегии, если какое-либо правило оценивается как "Permit", то результатом объединения правил должно быть "Permit". Если любое из правил оценивается как "Deny", а все другие правила

оцениваются как "NotApplicable", то стратегия должна оцениваться как "Deny". Другими словами, у "Permit" есть превосходство, независимо от результата оценки любого другого правила в этой стратегии. Если обнаруживается, что все правила "NotApplicable" к запросу о принятии решения, то стратегия должна оцениваться, как "NotApplicable".

Если произошла ошибка в процессе оценки цели или условия правила, в которой содержится значение эффекта "Permit", то процесс оценки должен продолжиться в поиске результата "Permit". Если больше ни одно правило не оценивается как "Permit", то стратегия должна оцениваться как "Indeterminate" с соответствующим статусом ошибки.

Если, по крайней мере, одно правило оценивается как "Deny", все другие правила, у которых нет ошибок оценки, оцениваются как "Deny" или "NotApplicable" и все правила, в которых нет ошибок оценки, содержащих значение эффекта "Deny", то стратегия должна оцениваться как "Deny".

Следующий псевдокод представляет стратегию оценки этого алгоритма объединения правил.

```
Decision permitOverridesRuleCombiningAlgorithm(Rule rule[])
{
   Boolean atLeastOneError = false;
   Boolean potentialPermit = false;
   Boolean atLeastOneDeny  = false;
   for( i=0 ; i < lengthOf(rule) ; i++ )
   {
          Decision decision = evaluate(rule[i]);
          if (decision == Deny)
          {
                 atLeastOneDeny = true;
                 continue;
          }
          if (decision == Permit)
          {
                 return Permit;
          }
          if (decision == NotApplicable)
          {
                 continue;
          }
          if (decision == Indeterminate)
          {
                 atLeastOneError = true;

                 if (effect(rule[i]) == Permit)
                 {
                        potentialPermit = true;
                 }
                 continue;
          }
   }
   if (potentialPermit)
   {
          return Indeterminate;
   }
   if (atLeastOneDeny)
   {
          return Deny;
   }
   if (atLeastOneError)
   {
          return Indeterminate;
   }
   return NotApplicable;
}
```

C.3.2 В этом пункте определяется алгоритм объединения стратегий "Permit-overrides" стратегического набора.

Во всем наборе стратегий в стратегическом наборе, если какая-либо стратегия оценивается как "Permit", то результатом объединения стратегии должно быть "Permit". Другими словами, у "Permit" есть преимущество, независимо от результата оценки любой другой стратегии в стратегическом наборе. Если обнаруживается, что все стратегии "NotApplicable" к запросу о принятии решения, то этот стратегический набор должен оцениваться как "NotApplicable".

Если произошла ошибка в процессе оценки цели стратегии или ссылка на стратегию считается неверной, или результатом оценки стратегии является "Indeterminate", то стратегический набор должен оцениваться как"Permit", "Deny".

Следующий псевдокод представляет стратегию оценки этого алгоритма объединения стратегии.

Если произошла ошибка в процессе оценки цели стратегии или ссылка на стратегию считается неверной, или результатом оценки стратегии является "Indeterminate", то стратегический набор должен оцениваться как "Indeterminate" с соответствующим статусом ошибки, при условии, что ни одна стратегия не оценивается как "Permit" или "Deny".

Следующий псевдокод представляет стратегию оценки этого алгоритма объединения стратегии.

```
Decision permitOverridesPolicyCombiningAlgorithm(Policy policy[])
{
   Boolean atLeastOneError = false;
   Boolean atLeastOneDeny = false;
   for( i=0 ; i < lengthOf(policy) ; i++ )
   {
          Decision decision = evaluate(policy[i]);
          if (decision == Deny)
          {
                atLeastOneDeny = true;
                continue;
          }
          if (decision == Permit)
          {
                return Permit;
          }
          if (decision == NotApplicable)
          {
                continue;
          }
          if (decision == Indeterminate)
          {
                atLeastOneError = true;
                continue;
          }
   }
   if (atLeastOneDeny)
   {
          return Deny;
   }
   if (atLeastOneError)
   {
          return Indeterminate;
   }
   return NotApplicable;
}
```

Обязательства отдельных стратегий должны объединяться, как описано в пункте 7.6.14.

C.4 Ordered-permit-overrides (Упорядоченное разрешение замен)

В следующем параграфе определяется алгоритм объединения правил "Ordered-permit-overrides" стратегии.

Поведение этого алгоритма идентично алгоритму объединения правил Permit-overrides с одним исключением. Порядок проведения оценки совокупности правил должен быть сопоставим с порядком, перечисленным в стратегии.

В следующем параграфе описывается алгоритм объединения стратегий набора "Ordered-permit-overrides" набора стратегии.

Поведение этого алгоритма идентично алгоритму объединения стратегий Permit-overrides с одним исключением. Порядок проведения оценки совокупности стратегий должен быть сопоставим с порядком, перечисленным в наборе стратегии.

C.5 First-applicable (Первый применим)

C.5.1 В этом пункте определяется алгоритм объединения правил "First-Applicable " стратегии.

Каждое правило должно оцениваться в том порядке, в каком оно перечислено в этой стратегии. Для конкретного правила, если цель сопоставима и условие оценивается как "True", то оценка стратегии должна быть остановлена и соответствующий эффект этого правила должен стать результатом оценки стратегии (т. е. "Permit" или "Deny"). Для конкретного правила, выбранного во время оценки, если цель оценивается как "False" или условие оценивается как "False", то должно оцениваться следующее по порядку правило. Если следующее по порядку правило отсутствует, то стратегия должна оцениваться, как "NotApplicable".

Если произошла ошибка в процессе оценки цели или условия правила, то оценка должна быть остановлена и стратегия должна оцениваться, как "Indeterminate", с соответствующим статусом ошибки.

Следующий псевдокод представляет стратегию оценки этого алгоритма объединения правил.

```
Decision firstApplicableEffectRuleCombiningAlgorithm(Rule rule[])
{
   for( i = 0 ; i < lengthOf(rule) ; i++ )
   {
          Decision decision = evaluate(rule[i]);
          if (decision == Deny)
          {
                return Deny;
          }
          if (decision == Permit)
          {
                return Permit;
          }
          if (decision == NotApplicable)
          {
                continue;
          }
          if (decision == Indeterminate)
          {
                return Indeterminate;
          }
   }
   return NotApplicable;
}
```

C.5.2 В этом пункте определяется алгоритм объединения стратегий "First-Applicable " стратегического набора.

Каждое правило должно оцениваться в том порядке, в каком оно перечислено в этом стратегическом наборе. Для конкретной стратегии, если цель оценивается как "True" и стратегия оценивается как определенное значение "Permit" или "Deny", то оценка должна быть остановлена и стратегический набор должен оцениваться, как значение эффекта этой стратегии. Для конкретной стратегии, если цель оценивается как "False" или стратегия оценивается как "NotApplicable", то должна оцениваться следующая по порядку стратегия. Если следующая по порядку стратегия отсутствует, то стратегический набор должен оцениваться как "NotApplicable".

Если произошла ошибка в процессе оценки цели или при оценке конкретной стратегии, ссылка на эту стратегию считается неверной, или сама стратегия оценивается как "Indeterminate", то оценка этого алгоритма объединения стратегий должна быть остановлена и стратегический набор должен оцениваться, как "Indeterminate", с соответствующим статусом ошибки.

Следующий псевдокод представляет стратегию оценки этого алгоритма объединения стратегий.

```
Decision firstApplicableEffectPolicyCombiningAlgorithm(Policy policy[])
{
  for( i = 0 ; i < lengthOf(policy) ; i++ )
  {
    Decision decision = evaluate(policy[i]);
    if(decision == Deny)
    {
      return Deny;
    }
    if(decision == Permit)
    {
      return Permit;
    }
    if (decision == NotApplicable)
    {
      continue;
    }
    if (decision == Indeterminate)
    {
      return Indeterminate;
    }
  }
  return NotApplicable;
}
```

Обязательства отдельных стратегий должны объединяться, как описано в пункте 7.6.14.

C.6 Only-one-applicable (Только один применим)

В этом пункте определяется алгоритм объединения стратегий "Only-one-applicable" стратегического набора.

Во всем наборе стратегий в стратегическом наборе, если ни одна стратегия не считается применимой на основании своей цели, то результатом алгоритма объединения стратегий должно быть "NotApplicable". Если более одной стратегии считается применимым на основании своей цели, то результатом алгоритма объединения стратегий должно быть "Indeterminate".

Если только одна стратегия считается применимой на основании оценки своей цели, то результатом алгоритма объединения стратегий должен быть результат оценки этой стратегии.

Если произошла ошибка в процессе оценки цели стратегии, или ссылка на стратегию считается неверной, или результатом оценки стратегии является "Indeterminate", то стратегический набор должен оцениваться как "Indeterminate", с соответствующим статусом ошибки.

Следующий псевдокод представляет стратегию оценки этого алгоритма объединения стратегий.

```
Decision onlyOneApplicablePolicyPolicyCombiningAlogrithm(Policy policy[])
{
 Boolean     atLeastOne   = false;
 Policy      selectedPolicy = null;
 ApplicableResult appResult;

 for ( i = 0; i < lengthOf(policy) ; i++ )
 {
   appResult = isApplicable(policy[I]);

   if ( appResult == Indeterminate )
   {
     return Indeterminate;
   }
   if( appResult == Applicable )
   {
     if ( atLeastOne )
     {
       return Indeterminate;
     }
     else
     {
       atLeastOne   = true;
       selectedPolicy = policy[i];
     }
   }
   if ( appResult == NotApplicable )
   {
     continue;
   }
 }
 if ( atLeastOne )
 {
   return evaluate(selectedPolicy);
 }
 else
 {
   return NotApplicable;
 }
}
```

Приложение D

Схема XACML

D.1 Схема контекста XACML

В этом пункте предоставляется схема контекста XACML.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:oasis:names:tc:xacml:2.0:context:schema:os"
xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
   <xs:import namespace="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
   schemaLocation="http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-
schema-os.xsd"/>
   <!--   -->
   <xs:element name="Request" type="xacml-context:RequestType"/>
   <xs:complexType name="RequestType">
         <xs:sequence>
               <xs:element ref="xacml-context:Subject" maxOccurs="unbounded"/>
               <xs:element ref="xacml-context:Resource" maxOccurs="unbounded"/>
               <xs:element ref="xacml-context:Action"/>
               <xs:element ref="xacml-context:Environment"/>
         </xs:sequence>
   </xs:complexType>
   <!-- -->
   <xs:element name="Response" type="xacml-context:ResponseType"/>
   <xs:complexType name="ResponseType">
         <xs:sequence>
               <xs:element ref="xacml-context:Result" maxOccurs="unbounded"/>
         </xs:sequence>
   </xs:complexType>
   <!-- -->
   <xs:element name="Subject" type="xacml-context:SubjectType"/>
   <xs:complexType name="SubjectType">
         <xs:sequence>
               <xs:element ref="xacml-context:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
         </xs:sequence>
         <xs:attribute name="SubjectCategory" type="xs:anyURI"
         default="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"/>
   </xs:complexType>
   <!-- -->
   <xs:element name="Resource" type="xacml-context:ResourceType"/>
   <xs:complexType name="ResourceType">
         <xs:sequence>
               <xs:element ref="xacml-context:ResourceContent" minOccurs="0"/>
               <xs:element ref="xacml-context:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
         </xs:sequence>
   </xs:complexType>
   <!-- -->
   <xs:element name="ResourceContent" type="xacml-context:ResourceContentType"/>
   <xs:complexType name="ResourceContentType" mixed="true">
         <xs:sequence>
               <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
         </xs:sequence>
         <xs:anyAttribute namespace="##any" processContents="lax"/>
   </xs:complexType>
   <!-- -->
   <xs:element name="Action" type="xacml-context:ActionType"/>
   <xs:complexType name="ActionType">
         <xs:sequence>
               <xs:element ref="xacml-context:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
         </xs:sequence>
```

```
   </xs:complexType>
   <!-- -->
   <xs:element name="Environment" type="xacml-context:EnvironmentType"/>
   <xs:complexType name="EnvironmentType">
          <xs:sequence>
                 <xs:element ref="xacml-context:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
   </xs:complexType>
   <!-- -->
   <xs:element name="Attribute" type="xacml-context:AttributeType"/>
   <xs:complexType name="AttributeType">
          <xs:sequence>
                 <xs:element ref="xacml-context:AttributeValue"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="AttributeId" type="xs:anyURI" use="required"/>
          <xs:attribute name="DataType" type="xs:anyURI" use="required"/>
          <xs:attribute name="Issuer" type="xs:string" use="optional"/>
   </xs:complexType>
   <!-- -->
   <xs:element name="AttributeValue" type="xacml-context:AttributeValueType"/>
   <xs:complexType name="AttributeValueType" mixed="true">
          <xs:sequence>
                 <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:anyAttribute namespace="##any" processContents="lax"/>
   </xs:complexType>
   <!-- -->
   <xs:element name="Result" type="xacml-context:ResultType"/>
   <xs:complexType name="ResultType">
          <xs:sequence>
                 <xs:element ref="xacml-context:Decision"/>
                 <xs:element ref="xacml-context:Status" minOccurs="0"/>
                 <xs:element ref="xacml:Obligations" minOccurs="0"/>
          </xs:sequence>
          <xs:attribute name="ResourceId" type="xs:string" use="optional"/>
   </xs:complexType>
   <!-- -->
   <xs:element name="Decision" type="xacml-context:DecisionType"/>
   <xs:simpleType name="DecisionType">
          <xs:restriction base="xs:string">
                 <xs:enumeration value="Permit"/>
                 <xs:enumeration value="Deny"/>
                 <xs:enumeration value="Indeterminate"/>
                 <xs:enumeration value="NotApplicable"/>
          </xs:restriction>
   </xs:simpleType>
   <!-- -->
   <xs:element name="Status" type="xacml-context:StatusType"/>
   <xs:complexType name="StatusType">
          <xs:sequence>
                 <xs:element ref="xacml-context:StatusCode"/>
                 <xs:element ref="xacml-context:StatusMessage" minOccurs="0"/>
                 <xs:element ref="xacml-context:StatusDetail" minOccurs="0"/>
          </xs:sequence>
   </xs:complexType>
   <!-- -->
   <xs:element name="StatusCode" type="xacml-context:StatusCodeType"/>
   <xs:complexType name="StatusCodeType">
          <xs:sequence>
                 <xs:element ref="xacml-context:StatusCode" minOccurs="0"/>
          </xs:sequence>
          <xs:attribute name="Value" type="xs:anyURI" use="required"/>
   </xs:complexType>
   <!-- -->
   <xs:element name="StatusMessage" type="xs:string"/>
   <!-- -->
   <xs:element name="StatusDetail" type="xacml-context:StatusDetailType"/>
   <xs:complexType name="StatusDetailType">
```

```
          <xs:sequence>
                <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
   </xs:complexType>
   <!-- -->
   <xs:element name="MissingAttributeDetail" type="xacml-
context:MissingAttributeDetailType"/>
   <xs:complexType name="MissingAttributeDetailType">
          <xs:sequence>
                <xs:element ref="xacml-context:AttributeValue" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="AttributeId" type="xs:anyURI" use="required"/>
          <xs:attribute name="DataType" type="xs:anyURI" use="required"/>
          <xs:attribute name="Issuer" type="xs:string" use="optional"/>
   </xs:complexType>
   <!-- -->
</xs:schema>
```

D.2 Схема стратегии

В этом пункте предоставляется схема стратегии XACML.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
elementFormDefault="qualified" attributeFormDefault="unqualified">
   <!-- -->
   <xs:element name="PolicySet" type="xacml:PolicySetType"/>
   <xs:complexType name="PolicySetType">
          <xs:sequence>
                <xs:element ref="xacml:Description" minOccurs="0"/>
                <xs:element ref="xacml:PolicySetDefaults" minOccurs="0"/>
                <xs:element ref="xacml:Target"/>
                <xs:choice minOccurs="0" maxOccurs="unbounded">
                      <xs:element ref="xacml:PolicySet"/>
                      <xs:element ref="xacml:Policy"/>
                      <xs:element ref="xacml:PolicySetIdReference"/>
                      <xs:element ref="xacml:PolicyIdReference"/>
                      <xs:element ref="xacml:CombinerParameters"/>
                      <xs:element ref="xacml:PolicyCombinerParameters"/>
                      <xs:element ref="xacml:PolicySetCombinerParameters"/>
                </xs:choice>
                <xs:element ref="xacml:Obligations" minOccurs="0"/>
          </xs:sequence>
          <xs:attribute name="PolicySetId" type="xs:anyURI" use="required"/>
          <xs:attribute name="Version" type="xacml:VersionType" default="1.0"/>
          <xs:attribute name="PolicyCombiningAlgId" type="xs:anyURI" use="required"/>
   </xs:complexType>
   <!-- -->
   <xs:element name="CombinerParameters" type="xacml:CombinerParametersType"/>
   <xs:complexType name="CombinerParametersType">
          <xs:sequence>
                <xs:element ref="xacml:CombinerParameter" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
   </xs:complexType>
   <!-- -->
   <xs:element name="CombinerParameter" type="xacml:CombinerParameterType"/>
   <xs:complexType name="CombinerParameterType">
          <xs:sequence>
                <xs:element ref="xacml:AttributeValue"/>
          </xs:sequence>
          <xs:attribute name="ParameterName" type="xs:string" use="required"/>
   </xs:complexType>
   <!-- -->
   <xs:element name="RuleCombinerParameters"
type="xacml:RuleCombinerParametersType"/>
```

```
   <xs:complexType name="RuleCombinerParametersType">
          <xs:complexContent>
                 <xs:extension base="xacml:CombinerParametersType">
                        <xs:attribute name="RuleIdRef" type="xs:string"
use="required"/>
                 </xs:extension>
          </xs:complexContent>
   </xs:complexType>
   <!-- -->
   <xs:element name="PolicyCombinerParameters"
type="xacml:PolicyCombinerParametersType"/>
   <xs:complexType name="PolicyCombinerParametersType">
          <xs:complexContent>
                 <xs:extension base="xacml:CombinerParametersType">
                        <xs:attribute name="PolicyIdRef" type="xs:anyURI"
use="required"/>
                 </xs:extension>
          </xs:complexContent>
   </xs:complexType>
   <!-- -->
   <xs:element name="PolicySetCombinerParameters"
type="xacml:PolicySetCombinerParametersType"/>
   <xs:complexType name="PolicySetCombinerParametersType">
          <xs:complexContent>
                 <xs:extension base="xacml:CombinerParametersType">
                        <xs:attribute name="PolicySetIdRef" type="xs:anyURI"
use="required"/>
                 </xs:extension>
          </xs:complexContent>
   </xs:complexType>
   <!-- -->
   <xs:element name="PolicySetIdReference" type="xacml:IdReferenceType"/>
   <xs:element name="PolicyIdReference" type="xacml:IdReferenceType"/>
   <!-- -->
   <xs:element name="PolicySetDefaults" type="xacml:DefaultsType"/>
   <xs:element name="PolicyDefaults" type="xacml:DefaultsType"/>
   <xs:complexType name="DefaultsType">
          <xs:sequence>
                 <xs:choice>
                        <xs:element ref="xacml:XPathVersion"/>
                 </xs:choice>
          </xs:sequence>
   </xs:complexType>
   <!-- -->
   <xs:element name="XPathVersion" type="xs:anyURI"/>
   <!-- -->
   <xs:complexType name="IdReferenceType">
          <xs:simpleContent>
                 <xs:extension base="xs:anyURI">
                        <xs:attribute name="Version" type="xacml:VersionMatchType"
use="optional"/>
                        <xs:attribute name="EarliestVersion"
type="xacml:VersionMatchType" use="optional"/>
                        <xs:attribute name="LatestVersion"
type="xacml:VersionMatchType" use="optional"/>
                 </xs:extension>
          </xs:simpleContent>
   </xs:complexType>
   <!-- -->
   <xs:simpleType name="VersionType">
          <xs:restriction base="xs:string">
                 <xs:pattern value="(\d+\.)*\d+"/>
          </xs:restriction>
   </xs:simpleType>
   <!-- -->
   <xs:simpleType name="VersionMatchType">
          <xs:restriction base="xs:string">
                 <xs:pattern value="((\d+|\*)\.)*(\d+|\*|\+)"/>
          </xs:restriction>
   </xs:simpleType>
   <!-- -->
```

```
<xs:element name="Policy" type="xacml:PolicyType"/>
<xs:complexType name="PolicyType">
       <xs:sequence>
              <xs:element ref="xacml:Description" minOccurs="0"/>
              <xs:element ref="xacml:PolicyDefaults" minOccurs="0"/>
              <xs:element ref="xacml:CombinerParameters" minOccurs="0"/>
              <xs:element ref="xacml:Target"/>
              <xs:choice maxOccurs="unbounded">
                     <xs:element ref="xacml:CombinerParameters" minOccurs="0"/>
                     <xs:element ref="xacml:RuleCombinerParameters" minOccurs="0"/>
                     <xs:element ref="xacml:VariableDefinition"/>
                     <xs:element ref="xacml:Rule"/>
              </xs:choice>
              <xs:element ref="xacml:Obligations" minOccurs="0"/>
       </xs:sequence>
       <xs:attribute name="PolicyId" type="xs:anyURI" use="required"/>
       <xs:attribute name="Version" type="xacml:VersionType" default="1.0"/>
       <xs:attribute name="RuleCombiningAlgId" type="xs:anyURI" use="required"/>
</xs:complexType>
<!-- -->
<xs:element name="Description" type="xs:string"/>
<!-- -->
<xs:element name="Rule" type="xacml:RuleType"/>
<xs:complexType name="RuleType">
       <xs:sequence>
              <xs:element ref="xacml:Description" minOccurs="0"/>
              <xs:element ref="xacml:Target" minOccurs="0"/>
              <xs:element ref="xacml:Condition" minOccurs="0"/>
       </xs:sequence>
       <xs:attribute name="RuleId" type="xs:string" use="required"/>
       <xs:attribute name="Effect" type="xacml:EffectType" use="required"/>
</xs:complexType>
<!-- -->
<xs:simpleType name="EffectType">
       <xs:restriction base="xs:string">
              <xs:enumeration value="Permit"/>
              <xs:enumeration value="Deny"/>
       </xs:restriction>
</xs:simpleType>
<!-- -->
<xs:element name="Target" type="xacml:TargetType"/>
<xs:complexType name="TargetType">
       <xs:sequence>
              <xs:element ref="xacml:Subjects" minOccurs="0"/>
              <xs:element ref="xacml:Resources" minOccurs="0"/>
              <xs:element ref="xacml:Actions" minOccurs="0"/>
              <xs:element ref="xacml:Environments" minOccurs="0"/>
       </xs:sequence>
</xs:complexType>
<!-- -->
<xs:element name="Subjects" type="xacml:SubjectsType"/>
<xs:complexType name="SubjectsType">
       <xs:sequence>
              <xs:element ref="xacml:Subject" maxOccurs="unbounded"/>
       </xs:sequence>
</xs:complexType>
<!-- -->
<xs:element name="Subject" type="xacml:SubjectType"/>
<xs:complexType name="SubjectType">
       <xs:sequence>
              <xs:element ref="xacml:SubjectMatch" maxOccurs="unbounded"/>
       </xs:sequence>
</xs:complexType>
<!-- -->
<xs:element name="Resources" type="xacml:ResourcesType"/>
<xs:complexType name="ResourcesType">
       <xs:sequence>
              <xs:element ref="xacml:Resource" maxOccurs="unbounded"/>
       </xs:sequence>
</xs:complexType>
<!-- -->
```

```
<xs:element name="Resource" type="xacml:ResourceType"/>
<xs:complexType name="ResourceType">
       <xs:sequence>
              <xs:element ref="xacml:ResourceMatch" maxOccurs="unbounded"/>
       </xs:sequence>
</xs:complexType>
<!-- -->
<xs:element name="Actions" type="xacml:ActionsType"/>
<xs:complexType name="ActionsType">
       <xs:sequence>
              <xs:element ref="xacml:Action" maxOccurs="unbounded"/>
       </xs:sequence>
</xs:complexType>
<!-- -->
<xs:element name="Action" type="xacml:ActionType"/>
<xs:complexType name="ActionType">
       <xs:sequence>
              <xs:element ref="xacml:ActionMatch" maxOccurs="unbounded"/>
       </xs:sequence>
</xs:complexType>
<!-- -->
<xs:element name="Environments" type="xacml:EnvironmentsType"/>
<xs:complexType name="EnvironmentsType">
       <xs:sequence>
              <xs:element ref="xacml:Environment" maxOccurs="unbounded"/>
       </xs:sequence>
</xs:complexType>
<!-- -->
<xs:element name="Environment" type="xacml:EnvironmentType"/>
<xs:complexType name="EnvironmentType">
       <xs:sequence>
              <xs:element ref="xacml:EnvironmentMatch" maxOccurs="unbounded"/>
       </xs:sequence>
</xs:complexType>
<!-- -->
<xs:element name="SubjectMatch" type="xacml:SubjectMatchType"/>
<xs:complexType name="SubjectMatchType">
       <xs:sequence>
              <xs:element ref="xacml:AttributeValue"/>
              <xs:choice>
                     <xs:element ref="xacml:SubjectAttributeDesignator"/>
                     <xs:element ref="xacml:AttributeSelector"/>
              </xs:choice>
       </xs:sequence>
       <xs:attribute name="MatchId" type="xs:anyURI" use="required"/>
</xs:complexType>
<!-- -->
<xs:element name="ResourceMatch" type="xacml:ResourceMatchType"/>
<xs:complexType name="ResourceMatchType">
       <xs:sequence>
              <xs:element ref="xacml:AttributeValue"/>
              <xs:choice>
                     <xs:element ref="xacml:ResourceAttributeDesignator"/>
                     <xs:element ref="xacml:AttributeSelector"/>
              </xs:choice>
       </xs:sequence>
       <xs:attribute name="MatchId" type="xs:anyURI" use="required"/>
</xs:complexType>
<!-- -->
<xs:element name="ActionMatch" type="xacml:ActionMatchType"/>
<xs:complexType name="ActionMatchType">
       <xs:sequence>
              <xs:element ref="xacml:AttributeValue"/>
              <xs:choice>
                     <xs:element ref="xacml:ActionAttributeDesignator"/>
                     <xs:element ref="xacml:AttributeSelector"/>
              </xs:choice>
       </xs:sequence>
       <xs:attribute name="MatchId" type="xs:anyURI" use="required"/>
</xs:complexType>
<!-- -->
```

```
   <xs:element name="EnvironmentMatch" type="xacml:EnvironmentMatchType"/>
   <xs:complexType name="EnvironmentMatchType">
          <xs:sequence>
                 <xs:element ref="xacml:AttributeValue"/>
                 <xs:choice>
                        <xs:element ref="xacml:EnvironmentAttributeDesignator"/>
                        <xs:element ref="xacml:AttributeSelector"/>
                 </xs:choice>
          </xs:sequence>
          <xs:attribute name="MatchId" type="xs:anyURI" use="required"/>
   </xs:complexType>
   <!-- -->
   <xs:element name="VariableDefinition" type="xacml:VariableDefinitionType"/>
   <xs:complexType name="VariableDefinitionType">
          <xs:sequence>
                 <xs:element ref="xacml:Expression"/>
          </xs:sequence>
          <xs:attribute name="VariableId" type="xs:string" use="required"/>
   </xs:complexType>
   <!-- -->
   <xs:element name="Expression" type="xacml:ExpressionType" abstract="true"/>
   <xs:complexType name="ExpressionType" abstract="true"/>
   <!-- -->
   <xs:element name="VariableReference"
   type="xacml:VariableReferenceType" substitutionGroup="xacml:Expression"/>
   <xs:complexType name="VariableReferenceType">
          <xs:complexContent>
                 <xs:extension base="xacml:ExpressionType">
                        <xs:attribute name="VariableId" type="xs:string"
use="required"/>
                 </xs:extension>
          </xs:complexContent>
   </xs:complexType>
   <!-- -->
   <xs:element name="AttributeSelector"
   type="xacml:AttributeSelectorType" substitutionGroup="xacml:Expression"/>
   <xs:complexType name="AttributeSelectorType">
          <xs:complexContent>
                 <xs:extension base="xacml:ExpressionType">
                        <xs:attribute name="RequestContextPath" type="xs:string"
use="required"/>
                        <xs:attribute name="DataType" type="xs:anyURI"
use="required"/>
                        <xs:attribute name="MustBePresent" type="xs:boolean"
use="optional" default="false"/>
                 </xs:extension>
          </xs:complexContent>
   </xs:complexType>
   <!-- -->
   <xs:element name="ResourceAttributeDesignator"
   type="xacml:AttributeDesignatorType" substitutionGroup="xacml:Expression"/>
   <xs:element name="ActionAttributeDesignator"
   type="xacml:AttributeDesignatorType" substitutionGroup="xacml:Expression"/>
   <xs:element name="EnvironmentAttributeDesignator"
   type="xacml:AttributeDesignatorType" substitutionGroup="xacml:Expression"/>
   <!-- -->
   <xs:complexType name="AttributeDesignatorType">
          <xs:complexContent>
                 <xs:extension base="xacml:ExpressionType">
                        <xs:attribute name="AttributeId" type="xs:anyURI"
use="required"/>
                        <xs:attribute name="DataType" type="xs:anyURI"
use="required"/>
                        <xs:attribute name="Issuer" type="xs:string" use="optional"/>
                        <xs:attribute name="MustBePresent" type="xs:boolean"
use="optional" default="false"/>
                 </xs:extension>
          </xs:complexContent>
   </xs:complexType>
   <!-- -->
   <xs:element name="SubjectAttributeDesignator"
```

```
   type="xacml:SubjectAttributeDesignatorType" substitutionGroup="xacml:Expression"/>
   <xs:complexType name="SubjectAttributeDesignatorType">
          <xs:complexContent>
                 <xs:extension base="xacml:AttributeDesignatorType">
                        <xs:attribute name="SubjectCategory" type="xs:anyURI"
                        use="optional" default="urn:oasis:names:tc:xacml:1.0:subject-
category:access-subject"/>
                 </xs:extension>
          </xs:complexContent>
   </xs:complexType>
   <!-- -->
   <xs:element name="AttributeValue" type="xacml:AttributeValueType"
          substitutionGroup="xacml:Expression"/>
   <xs:complexType name="AttributeValueType" mixed="true">
          <xs:complexContent mixed="true">
                 <xs:extension base="xacml:ExpressionType">
                        <xs:sequence>
                               <xs:any namespace="##any" processContents="lax"
minOccurs="0"
                                      maxOccurs="unbounded"/>
                        </xs:sequence>
                        <xs:attribute name="DataType" type="xs:anyURI"
use="required"/>
                        <xs:anyAttribute namespace="##any" processContents="lax"/>
                 </xs:extension>
          </xs:complexContent>
   </xs:complexType>
   <!-- -->
   <xs:element name="Function" type="xacml:FunctionType"
substitutionGroup="xacml:Expression"/>
   <xs:complexType name="FunctionType">
          <xs:complexContent>
                 <xs:extension base="xacml:ExpressionType">
                        <xs:attribute name="FunctionId" type="xs:anyURI"
use="required"/>
                 </xs:extension>
          </xs:complexContent>
   </xs:complexType>
   <!-- -->
   <xs:element name="Condition" type="xacml:ConditionType"/>
   <xs:complexType name="ConditionType">
          <xs:sequence>
                 <xs:element ref="xacml:Expression"/>
          </xs:sequence>
   </xs:complexType>
   <!-- -->
   <xs:element name="Apply" type="xacml:ApplyType"
substitutionGroup="xacml:Expression"/>
   <xs:complexType name="ApplyType">
          <xs:complexContent>
                 <xs:extension base="xacml:ExpressionType">
                        <xs:sequence>
                               <xs:element ref="xacml:Expression" minOccurs="0"
maxOccurs="unbounded"/>
                        </xs:sequence>
                        <xs:attribute name="FunctionId" type="xs:anyURI"
use="required"/>
                 </xs:extension>
          </xs:complexContent>
   </xs:complexType>
   <!-- -->
   <xs:element name="Obligations" type="xacml:ObligationsType"/>
   <xs:complexType name="ObligationsType">
          <xs:sequence>
                 <xs:element ref="xacml:Obligation" maxOccurs="unbounded"/>
          </xs:sequence>
   </xs:complexType>
   <!-- -->
   <xs:element name="Obligation" type="xacml:ObligationType"/>
   <xs:complexType name="ObligationType">
          <xs:sequence>
```

```
                <xs:element ref="xacml:AttributeAssignment" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ObligationId" type="xs:anyURI" use="required"/>
          <xs:attribute name="FulfillOn" type="xacml:EffectType" use="required"/>
    </xs:complexType>
    <!-- -->
    <xs:element name="AttributeAssignment" type="xacml:AttributeAssignmentType"/>
    <xs:complexType name="AttributeAssignmentType" mixed="true">
          <xs:complexContent mixed="true">
                <xs:extension base="xacml:AttributeValueType">
                      <xs:attribute name="AttributeId" type="xs:anyURI"
use="required"/>
                </xs:extension>
          </xs:complexContent>
    </xs:complexType>
    <!-- -->
</xs:schema>
```

D.3 Схема протокола SAML XACML

В этом пункте предоставляется схема протокола SAML XACML.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
  targetNamespace="urn:oasis:xacml:2.0:saml:protocol:schema:os"
  xmlns:xs="http://www.23.org/2001/XMLSchema"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
  xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os"
  xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
  elementFormDefault="unqualified"
  attributeFormDefault="unqualified"
  blockDefault="substitution"
  version="2.0">
 <xs:import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
   schemaLocation="http://www.oasis-
open.org/committees/tc_home.php?wg_abbrev=security"/>
 <xs:import namespace="urn:oasis:names:tc:SAML:2.0:protocol"
   schemaLocation="http://www.oasis-
open.org/committees/tc_home.php?wg_abbrev=security"/>
 <xs:import namespace="urn:oasis:names:tc:xacml:2.0:context:schema:os"
   schemaLocation="http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-
context-schema-os.xsd"/>
 <xs:import namespace="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
   schemaLocation="http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-
policy-schema-os.xsd"/>
 <xs:annotation>
  <xs:documentation>
    Document identifier: access_control-xacml-2.0-saml-protocol-schema-os.xsd
    Location: http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-saml-
protocol-schema-os.xsd
  </xs:documentation>
 </xs:annotation>
 <!--  -->
 <xs:element name="XACMLAuthzDecisionQuery"
      type="XACMLAuthzDecisionQueryType"/>
 <xs:complexType name="XACMLAuthzDecisionQueryType">
  <xs:complexContent>
   <xs:extension base="samlp:RequestAbstractType">
    <xs:sequence>
     <xs:element ref="xacml-context:Request"/>
    </xs:sequence>
    <xs:attribute name="InputContextOnly"
           type="boolean"
           use="optional"
           default="false"/>
    <xs:attribute name="ReturnContext"
           type="boolean"
```

```
          use="optional"
          default="false"/>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
 <!--  -->
 <xs:element name="XACMLPolicyQuery"
      type="XACMLPolicyQueryType"/>
 <xs:complexType name="XACMLPolicyQueryType">
  <xs:complexContent>
   <xs:extension base="samlp:RequestAbstractType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">>
     <xs:element ref="xacml-context:Request"/>
     <xs:element ref="xacml:Target"/>
     <xs:element ref="xacml:PolicySetIdReference"/>
     <xs:element ref="xacml:PolicyIdReference"/>
    </xs:choice>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
</schema>
```

D.4 Схема утверждения SAML XACML

В этом пункте предоставляется схема утверждения SAML XACML.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
  targetNamespace="urn:oasis:xacml:2.0:saml:assertion:schema:os"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:xs="http://www.23.org/2001/XMLSchema"
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
  xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os"
  xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
  elementFormDefault="unqualified"
  attributeFormDefault="unqualified"
  blockDefault="substitution"
  version="2.0">
 <xs:import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
   schemaLocation="http://www.oasis-
open.org/committees/tc_home.php?wg_abbrev=security"/>
 <xs:import namespace="urn:oasis:names:tc:SAML:2.0:protocol"
   schemaLocation="http://www.oasis-
open.org/committees/tc_home.php?wg_abbrev=security"/>
 <xs:import namespace="urn:oasis:names:tc:xacml:2.0:context:schema:os"
   schemaLocation="http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-
context-schema-os.xsd"/>
 <xs:import namespace="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
   schemaLocation="http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-
policy-schema-os.xsd"/>
 <xs:annotation>
  <xs:documentation>
    Document identifier: access_control-xacml-2.0-saml-assertion-schema-cd-02.xsd
    Location: http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-saml-
assertion-schema-cd-os.xsd
  </xs:documentation>
 </xs:annotation>
 <!--  -->
 <xs:element name="XACMLAuthzDecisionStatement"
      type="XACMLAuthzDecisionStatementType"/>
 <xs:complexType name="XACMLAuthzDecisionStatementType">
  <xs:complexContent>
   <xs:extension base="samlp:StatementAbstractType">
    <xs:sequence>
     <xs:element ref="xacml-context:Response"/>
     <xs:element ref="xacml-context:Request" MinOccurs="0"/>
    </xs:sequence>
   </xs:extension>
  </xs:complexContent>
```

```
 </xs:complexType>
 <!--  -->
 <xs:element name="XACMLPolicyStatement"
      type="XACMLPolicyStatementType"/>
 <xs:complexType name="XACMLPolicyStatementType">
  <xs:complexContent>
   <xs:extension base="samlp:StatementAbstractType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">>
     <xs:element ref="xacml:Policy"/>
     <xs:element ref="xacml:PolicySet"/>
    </xs:choice>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
</schema>
```

Дополнение I

Соображения безопасности

В этом дополнении идентифицированы возможные сценарии дискредитации безопасности и секретности, которые должны быть приняты во внимание при реализации систем на основе XACML. Решение о том, являются ли эти сценарии дискредитации реальными в определенной среде и выбор подходящих мер безопасности оставлен на рассмотрение конструктора.

I.1 Модель угрозы

Мы здесь предполагаем, что враг имеет доступ к каналу связи между действующими субъектами XACML и может интерпретировать, вставлять, удалять и изменять сообщения или части сообщений.

Дополнительно, действующий субъект может злонамеренно использовать информацию из предшествующего сообщения в последующих транзакциях. Далее предполагается, что правила и стратегии надежны до степени надежности действующих субъектов, которые их создали и используют. Таким образом, на действующий субъект возлагается установление должного доверия к другим действующим субъектам, на которых он рассчитывает. Механизмы установления доверия выходят за рамки данной Рекомендации.

Сообщения, которые передаются между действующими субъектами в модели XACML восприимчивы к атакам со стороны злонамеренных третьих лиц. Другими моментами уязвимости являются PEP, PDP и PAP. Так как некоторые из этих объектов не находятся строго в рамках данной Рекомендации, то их дискредитация может привести к дискредитации контроля доступа, осуществляемого с помощью PEP.

Следует заметить, что существуют другие компоненты распределенной системы, которые могут быть дискредитированы, такие как операционная система и система доменных имен (DNS), которые находятся за рамками этого обсуждения моделей угроз.

В следующих пунктах подробно изложены конкретные сценарии дискредитации, которые могут относиться к системе XACML.

I.1.1 Несанкционированное разглашение

В XACML не установлены какие-либо неотъемлемые механизмы для защиты конфиденциальности сообщений, которыми обмениваются действующие субъекты. Таким образом, враг может наблюдать за сообщениями во время передачи. При известных стратегиях безопасности, разглашение этой информации является нарушением. Разглашение атрибутов или типов запросов о принятии решения, которые субъект представляет на рассмотрение, может быть нарушением стратегии секретности. В невоенной отрасли последствия несанкционированного разглашения личных данных могут находиться в диапазоне от затруднений для хранителя до заключения в тюрьму и больших штрафов, в случае медицинских или финансовых данных.

О несанкционированном разглашении сообщается с помощью мер безопасности конфиденциальности.

I.1.2 Воспроизведение сообщения

Атака воспроизведения сообщения – это такая атака, при которой враг записывает и воспроизводит настоящие сообщения между действующими субъектами XACML. Эта атака может привести к отказу в обслуживании, использованию устаревшей информации или выдаче себя за другого человека.

Предотвращение атак воспроизведения требует использования мер безопасности по обновлению сообщений.

Заметим, что шифрование сообщения не ослабляет атаку воспроизведения, так как сообщение просто воспроизводится и его понимание врагом не имеет значения.

I.1.3 Введение сообщения

Атака введения сообщений – это такая атака, при которой враг вводит сообщения в последовательность сообщений между действующими субъектами XACML.

Решением для атак введения сообщений является использование взаимной аутентификации и мер безопасности по сохранению целостности последовательности сообщений между действующими субъектами. Следует заметить, что использование только взаимной аутентификации SSL не является достаточным. Она только подтверждает, что другая сторона – это та, которая идентифицируется субъектом сертификата X.509. Для результативности необходимо подтвердить, что этому субъекту сертификата разрешено отправлять сообщение.

I.1.4 Удаление сообщения

Атака удаления сообщения – это такая атака, при которой враг удаляет сообщения в последовательности сообщений между действующими субъектами XACML. Удаление сообщения может привести к отказу в обслуживании. Однако, надлежащим образом разработанная, система XACML не должна отдавать неверные решения об авторизации в результате атаки удаления сообщения.

Решением для атаки удаления сообщения является использование мер безопасности по сохранению целостности последовательности сообщений между действующими субъектами.

I.1.5 Изменение сообщения

Если враг может перехватить сообщение и изменить его содержание, то он может изменить решение об авторизации. Меры безопасности по сохранению целостности сообщения могут предотвратить успешную атаку по изменению сообщения.

I.1.6 Результаты NotApplicable

Результат "NotApplicable" означает, что PDP не может определить местоположение стратегии, цель которой сопоставима с информацией в запросе о принятии решения. В общем случае настоятельно рекомендуется использование стратегии эффекта "Deny" таким образом, если бы PDP возвратил "NotApplicable", то вместо этого возвращался бы результат "Deny".

В некоторых моделях безопасности, таких как те, которые могут быть обнаружены во многих Web Servers, решение об авторизации "NotApplicable" считается эквивалентом решения "Permit". Существуют специфические соображения безопасности, которые нужно учитывать в этом случае, чтобы сохранить безопасность. Они объясняются в следующих параграфах.

Если "NotApplicable" считается эквивалентом "Permit", то жизненно важно, чтобы алгоритмы сопоставления для сопоставления элементов в запросе о принятии решения, используемые этой стратегии, точно были бы выровнены с синтаксисом данных, используемом теми приложениями, которые будут представлять на рассмотрение запрос о принятии решения. Неудача при сопоставлении приведет к результату "NotApplicable" и он будет воспринят, как "Permit". Поэтому, непреднамеренный сбой при сопоставлении может разрешить непреднамеренный доступ.

Коммерческие http респонденты разрешают многообразие синтаксических выражений, с которыми следует обращаться, как с эквивалентами. "%" может использоваться для представления символов с помощью шестнадцатеричного значения. Путь URL "/../" предоставляет множество способов установления одного и того же значения. Может быть разрешено множество наборов символов и, в некоторых случаях, один и тот же печатный символ может быть представлен разными двоичными значениями. Непреднамеренный доступ может быть разрешен, если алгоритм сопоставления, используемый стратегией, недостаточно сложный для того, чтобы улавливать эти различия.

Безопасным может быть использование "NotApplicable" в качестве эквивалента "Permit" только в закрытой среде, в которой можно гарантировать, что все приложения, формулирующие запрос о принятии решения, используют именно тот синтаксис, который ожидают стратегии. В более открытых средах, в которых запросы о принятии решения могут быть получены от приложений, использующих любой законный синтаксис, настоятельно рекомендуется, чтобы "NotApplicable" не воспринималось, как "Permit", если правила сопоставления не были разработаны очень тщательно, для сопоставления всех возможных применяемых входных данных, независимо от вариаций синтаксиса или типа. PEP должен запрещать доступ, если он не получает явное решение об авторизации "Permit".

I.1.7 Правила отрицания

Правило отрицания – это такое правило, которое основано на предикате: не являющееся "True". Если обращаться с ними неосторожно, то правила отрицания могут привести к нарушению стратегии, поэтому некоторые органы не рекомендуют их использовать. Однако в некоторых случаях правила отрицания могут быть чрезвычайно эффективными, поэтому в XACML сделан выбор: использовать их. Тем не менее рекомендуется использовать их с осторожностью и, по возможности, избегать их использования.

Распространенным использованием правил отрицания является запрет доступа для отдельного представителя или подгруппы, если их членство в более крупной подгруппе, иначе, разрешило бы им доступ. Например, мы можем захотеть написать правило, которое позволит всем вице президентам видеть неопубликованные финансовые данные, за исключением Джо, который является лишь протокольным вице президентом и может быть неосмотрительным в своих сообщениях. Если у нас имеется полный контроль над управлением атрибутами субъекта, наилучшим подходом было бы определение "вице президента" и "протокольного вице президента", как различных групп, а затем, в соответствии с этим, определение правил. Однако в некоторых средах такой подход может быть невыполнимым. (Стоит отметить, между прочим, что вообще говоря, ссылка на отдельных представителей в правилах плохо масштабируется. В общем случае, предпочтительнее совместно используемые атрибуты.)

Если обращаться с правилами отрицания неосторожно, то это может привести к нарушению стратегии в двух распространенных случаях. Это случаи, когда скрываются атрибуты и когда изменяется базовая группа. Примером скрытия атрибутов будет случай, если у нас имеется стратегия, доступ при которой разрешен, если

субъектом не является кредитный риск. Если возможна такая ситуация, при которой атрибут, являющийся кредитным риском, может быть неизвестен пункту PDP по какой-либо причине, то результатом может быть несанкционированный доступ. В некоторых средах субъект может иметь возможность скрывать публикацию атрибутов с помощью приложения контроля секретности, или сервер, или репозиторий, содержащий эту информацию, может быть недоступен по случайным или преднамеренным причинам.

Примером изменения базовой группы мог бы послужить случай существования стратегии, при которой в инженерно-техническом отделе изменяется код источника математического обеспечения, за исключением секретарей. Предположим теперь, что этот отдел должен был объединиться с другим инженерно-техническим отделом и есть намерение сохранить ту же самую стратегию. Однако в новый отдел также включены отдельные представители, идентифицированные, как помощники по административной работе, с которыми надо обращаться так же, как и с секретарями. Если стратегия не поменяется, то им должно быть дано ненамеренное разрешение изменить код источника математического обеспечения. Проблем такого типа можно легко избежать, если один отдельный представитель управляет всеми стратегиями, но если управление распределено, что позволяется языком XACML, то нужно иметь явную защиту для ситуации такого типа.

I.2 Меры безопасности

I.2.1 Аутентификация

Аутентификация предоставляет средства для одной стороны в транзакции определить идентичность другой стороны в этой транзакции. Аутентификация может односторонняя или двусторонняя.

Если задана высокая чувствительность для систем контроля доступа, то для PEP важно аутентифицировать идентичность PDP, которому он отправляет запросы о принятии решения. Иначе, существует риск того, что враг может предоставить фальшивые или неверные решения об авторизации, что приведет к нарушению стратегии.

В равной степени важно для PDP аутентифицировать идентичность PEP и оценить уровень доверия, для того чтобы определить какие, если имеются, чувствительные данные должны быть переданы. Нужно иметь в виду, что даже простые ответы "Permit" или "Deny" могут быть использованы, если врагу было бы разрешено отправлять неограниченные запросы в PDP.

Много разных методов можно использовать для предоставления аутентификации, таких как совместное размещение кода, частная сеть, VPN (виртуальная частная сеть) или цифровые подписи. Аутентификация может также выполняться, как часть протокола связи, используемого для обмена контекстами. В этом случае аутентификация может выполняться либо на уровне сообщений, либо на уровне сеансов.

I.2.2 Управление стратегией

Если содержание стратегий раскрывается за пределами системы контроля доступа, то потенциальные субъекты могут использовать эту информацию, для того чтобы определить, как добиться несанкционированного доступа.

Для предотвращения этой угрозы, репозиторий, использующийся для хранения стратегий, может сам потребовать контроль доступа. Дополнительно, элемент `<Status>` должен использоваться для возвращения значений пропущенных атрибутов только, если раскрытие идентичностей этих атрибутов не будет дискредитировать безопасность.

I.2.3 Конфиденциальность

Механизмы конфиденциальности гарантируют, что содержание сообщения может быть считано только желательными получателями, а не любым другим, кому случайно попадет это сообщение при его передаче. Существуют две области, в которых должна рассматриваться конфиденциальность: одна область – это конфиденциальность во время передачи; другая область – это конфиденциальность внутри элемента `<Policy>`.

I.2.3.1 Конфиденциальность связи

В некоторых средах считается хорошей практикой обращаться со всеми данными внутри системы контроля доступа, как с конфиденциальными. В других средах к стратегиям может быть открытый доступ для распределения, проверки и аудита можно сделать могут быть. Идея сохранения информации о стратегии в секрете заключается в том, чтобы затруднить получение врагом сведений о том, какие шаги могут быть достаточными для получения несанкционированного доступа. Независимо от выбранного подхода, безопасность системы контроля доступа не должна зависеть от секретности стратегии.

Любые соображения безопасности, связанные с передачей или обменом элементов `<Policy>` XACML выходят за рамки стандарта языка XACML. В то время, как часто бывает важно гарантировать сохранение целостности и конфиденциальности элементов `<Policy>`, при их обмене между двумя сторонами, определение подходящих механизмов для своих сред отдается на усмотрение конструкторов.

Конфиденциальность связи может быть предоставлена механизмом конфиденциальности, таким как SSL. Используя схему связи пункта с пунктом, SSL может привести к другим уязвимостям, если одна из конечных точек дискредитирована.

I.2.3.2 Конфиденциальность уровня заявления

В некоторых случаях, в реализации может понадобиться зашифровать только части элемента `<Policy>` XACML `<Policy>`.

W3C Encryption:2002 может использоваться для шифрования всего или частей документа XML. W3C Encryption:2002 рекомендован для использования с XACML.

Безусловным должно быть то, что если репозиторий используется для упрощения передачи ясного (т. е. незашифрованного) текста стратегии между PAP и PDP, то защищенный репозиторий должен использоваться для хранения этих чувствительных данных.

I.2.4 Целостность стратегии

Стратегия XACML, используемая пунктом PDP для оценки контекста запроса, является основой всей системы. Таким образом, сохранение целостности является существенным. Есть два аспекта сохранения целостности стратегии. Один аспект для гарантирования того, что элементы `<Policy>` не были изменены, с тех пор, как они были первоначально созданы с помощью PAP. Другой аспект для гарантирования того, что элементы `<Policy>` не были вставлены или удалены из набора стратегий.

Во многих случаях, можно достичь обоих аспектов с помощью гарантирования целостности действующих субъектов и механизмов сеансового уровня реализаций для защиты связи между действующими субъектами. Выбор подходящих механизмов оставлен на усмотрение конструкторов. Однако, если стратегия распределена между организациями, которые должны действовать в более позднее время, или если стратегия путешествует вместе с защищаемым ресурсом, то было бы полезно подписать эту стратегию. В этих случаях рекомендуется использовать синтаксис подписи XML и стандарт обработки из W3C с языком XACML.

Цифровые подписи должны использоваться только для гарантирования целостности заявлений. Цифровые подписи не должны использоваться в качестве метода выбора или оценки стратегии. То есть, PDP не должен запрашивать стратегию, основанную на том, кто подписал ее, или на том была она подписана или нет (так как такое основание для выбора, само по себе, есть вопрос стратегии). Однако, PDP должен проверить, что ключ, используемый для подписания стратегии, является ключом, которым управляет подразумеваемая запрашивающая сторона стратегии. Средства для выполнения этого зависят от выбранной конкретной технологии подписи и выходят за рамки данной Рекомендации.

I.2.5 Идентификаторы стратегии

Так как на стратегии можно ссылаться с помощью их идентификаторов, то сферой ответственности PAP является гарантирование их уникальности. Путаница с идентификаторами может привести к неправильной идентификации применяемой стратегии. В данной Рекомендации умалчивается о том, должен ли PAP создавать новый идентификатор, если стратегия изменяется, или можно использовать тот же самый идентификатор в измененной стратегии. Это вопрос практики административного управления. Однако, нужно действовать с осторожностью в любом случае. Если идентификатором пользуются многократно, то есть опасность, что другие стратегии или набор стратегий, которые на нее ссылаются, могут попасть под воздействие врага. Наоборот, если используется новый идентификатор, то эти новые стратегии могут продолжать пользоваться предыдущей стратегией, если она не удалена. В любом случае, результат может оказаться не тем, на который рассчитывает администратор.

I.2.6 Модель доверия

При обсуждения мер безопасности аутентификации, целостности и конфиденциальности необходимо принять базовую модель доверия: каким образом действующий субъект может убедиться в том, что заданный ключ однозначно связан с конкретным, идентифицированным действующим субъектом таким образом, что этот ключ может быть использован для шифрования данных для этого действующего субъекта или проверить подписи (или другие структуры целостности) от этого действующего субъекта? Существует много разных типов модели доверия, включая строгие иерархии, распределенные органы, распределенная информационная сеть, Web (Глобальная сеть), мост и т. п.

Стоит рассмотреть взаимосвязи между различными действующими субъектами системы контроля доступа, исходя из существования взаимозависимостей или их отсутствия.

- Ни один из объектов системы авторизации не зависит от PEP. Они могут собирать данные, поступающие от них, например данные аутентификации, но сами отвечают за их проверку.
- Правильная работа этой системы зависит от способности PEP фактически осуществлять решения стратегии.
- PEP зависит от того, насколько правильно PDP оценивает стратегии. Это, в свою очередь, подразумевает, что PDP обеспечивается правильными входными данными.А кроме этого PDP не зависит от PEP.
- PDP зависит от снабжения подходящими стратегиями со стороны PAP. Пункт PAP не зависит от других компонентов.

I.2.7 Секретность

Важно знать, что транзакции, которые происходят с учетом контроля доступа, могут открывать сведения не подлежащие огласке о действующих субъектах. Например, если стратегия XACML заявляет, что некоторые данные могут быть считаны только субъектами со статусом "Золотая карточка члена организации", то любая транзакция, в которой субъекту разрешается доступ к этим данным, дает утечку информации к врагу о статусе этого субъекта. Соображения секретности могут, таким образом, привести к шифрованию и/или к требованиям контроля доступа для самих объектов стратегии XACML, окружающих ее осуществление: каналов, защищенных в плане конфиденциальности, для сообщений протоколов запросов/ответов, защита атрибутов субъектов при их хранении и передаче и т. п.

Выбор и использование подходящих механизмов секретности для заданной среды выходит за рамки XACML. Решение относительно того, как и где развертывать такие механизмы оставлено на усмотрение конструкторов, связанных с этой средой.

Дополнение II

Примеры XACML

В этом дополнении содержится два примера использования XACML с целью иллюстрации. Первым является сравнительно простой пример для иллюстрации использования цели, контекста, функций сопоставления и атрибутов субъекта. Во втором примере дополнительно иллюстрируется использование алгоритма объединения правил, условия и обязательства.

II.1 Пример один

В этом пункте содержится первый пример.

II.1.1 Пример стратегии

Предположим, что корпорация, под названием Medi Corp (идентифицируемая по своему доменному имени: med.example.com) имеет стратегию контроля доступа, которая заявляет на английском языке:

Любому пользователю с именем электронной почты в пространстве имен "med.example.com" разрешено выполнять любое действие над любым ресурсом.

Стратегия XACML состоит из информации заголовка, необязательного текстового описания этой стратегии, **цели**, одного или более **правил** и необязательного набора **обязательств**.

```
[a02]  <?xml version="1.0" encoding="UTF-8"?>
[a03]  <Policy
[a04]  xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
[a05]  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
[a06]  xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-
os.xsd"
[a07]  PolicyId="urn:oasis:names:tc:example:SimplePolicy1"
[a08]  RuleCombiningAlgId="identifier:rule-combining-algorithm:deny-
overrides">
[a09]   <Description>
[a10]   Medi Corp access control policy
[a11]   </Description>
[a12]   <Target/>
[a13]   <Rule
[a14]   RuleId= "urn:oasis:names:tc:xacml:2.0:example:SimpleRule1"
[a15]   Effect="Permit">
[a16]   <Description>
[a17]    Any subject with an e-mail name in the med.example.com domain
[a18]    can perform any action on any resource.
[a19]   </Description>
[a20]   <Target>
[a21]    <Subjects>
[a22]    <Subject>
[a23]     <SubjectMatch
[a24]     MatchId="urn:oasis:names:tc:xacml:1.0:function:rfc822Name-match">
[a25]     <AttributeValue
[a26]     DataType="http://www.w3.org/2001/XMLSchema#string">
[a27]      med.example.com
[a28]     </AttributeValue>
[a29]     <SubjectAttributeDesignator
[a30]     AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
[a31]     DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name"/>
[a32]     </SubjectMatch>
[a33]    </Subject>
[a34]    </Subjects>
[a35]   </Target>
[a36]   </Rule>
[a37]  </Policy>
```

[a02] является стандартным маркером документа XML, показывающим, какая версия XML используется и какое шифрование символов.

В [a03] вводится сама стратегия XACML.

[a04] – [a05] являются описаниями пространства имен XML.

В [a04] задается URN для схемы стратегии XACML.

В [a07] присваивается имя этому экземпляру стратегии. Имя стратегии должно быть уникальным для заданного PDP, таким образом, чтобы не было неопределенности, если на одну стратегию ссылаются из другой стратегии. Атрибут версии пропущен, поэтому взято его значение по умолчанию "1.0".

В [a08] устанавливается алгоритм, который будет использоваться, чтобы разрешить результаты различных правил, которые могут быть в этой стратегии. Алгоритм объединения правил deny-overrides, установленный здесь, говорит о том, что если любое правило оценивается как "Deny", то стратегия должна возвращать "Deny". Если все правила оцениваются как "Permit", то эта стратегия должна возвращать "Permit". Алгоритм объединения правил, который полностью описан в Приложении C, также говорит о том, что делать, если при оценке какого-либо правила произошла ошибка, и что делать с правилами, которые не применяются к конкретному запросу о принятии решения.

В [a09] – [a11] предоставляется текстовое описание этой стратегии. Это описание является необязательным.

В [a12] описываются запросы о принятии решения, к которым применяется эта стратегия. Если субъект, ресурс, действие или среда в запросе о принятии решения не сопоставимы со значениями, установленными в цели стратегии, то нет необходимости оценивать оставшуюся часть стратегии. Этот сектор цели полезен для создания индекса для набора стратегий. В этом простом примере сектор цели говорит о том, что эта стратегия применима к любому запросу о принятии решения.

В [a13] вводится одно и только одно правило в этой простой стратегии.

В [a14] устанавливается идентификатор для этого правила. Так же, как и для стратегий, для каждого правила должен быть уникальный идентификатор (по крайней мере, уникальный для любого PDP, который будет использоваться в стратегии).

В [a15] говорится о том, какой эффект будет у этого правила, если это правило будет оценено, как "True". У правил может быть эффект либо "Permit", либо "Deny". В этом случае, если правило исполнено, то оно будет оценено, как "Permit", и это означает, что поскольку это правило исполнено, то запрашиваемый доступ должен быть получен. Если правило оценивается как "False", то возвращается результат "NotApplicable". Если при оценке правила происходит ошибка, то правило возвращает результат "Indeterminate". Как было упомянуто выше, алгоритм объединения правил для этой стратегии устанавливает, каким образом различные значения правила объединяются в единое значение стратегии.

В [a16] – [a19] предоставляется текстовое описание этого правила. Это описание является необязательным.

В [a20] вводится цель этого правила. Как описано выше по поводу цели стратегии, цель правила описывает запросы о принятии решения, к которым эти правила применяются. Если субъект, ресурс, действие или среда в запросе о принятии решения не сопоставимы со значениями, установленными в цели правила, то нет необходимости оценивать оставшуюся часть стратегии и для оценки правила возвращается значение "NotApplicable".

Цель правила сходна с самой целью стратегии, но есть одно важное отличие. В [a23] – [a32] разъясняется конкретное значение, с которым должен быть сопоставим субъект в этом запросе о принятии решения. В элементе `<SubjectMatch>` устанавливается функция сопоставления в атрибуте `MatchId`, буквенное значение "med.example.com" и указатель к конкретному атрибуту субъекта в контексте запроса посредством элемента `<SubjectAttributeDesignator>`. Функция сопоставления будет использоваться для сравнения буквенного значения со значением атрибута субъекта. Только если сопоставление возвратит "True", это правило будет применимо к конкретному запросу о принятии решения. Если сопоставление возвращает "False", то это правило возвратит "NotApplicable".

В [a36] правило закрывается. В этом правиле вся работа выполняется в элементе `<Target>`. В более сложных правилах за `<Target>` может следовать `<Condition>` (который тоже может быть набором условий, которые объединяются вместе с помощью логических AND или OR).

В [a37] стратегия закрывается. Как упомянуто выше, у этой стратегии есть только одно правило, но у более сложных стратегий может быть любое количество правил.

II.1.2 Пример контекста запроса

Давайте рассмотрим гипотетический запрос о принятии решения, который может быть представлен в PDP, выполняющий стратегию, описанную выше. В английском языке запрос доступа, который создает запрос о принятии решения, может быть заявлен следующим образом:

Bart Simpson, с именем электронной почты "bs@simpsons.com", хочет считать свою медицинскую карту в Medi Corp.

В XACML информация в запросе о принятии решения форматируется в заявление контекста запроса, которое выглядит следующим образом:

```
[a38]  <?xml version="1.0" encoding="UTF-8"?>
[a39]  <Request xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
[a40]  xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:context:schema:os
http://docs.oasis-
open.org/xacml/access_control-xacml-2.0-context-schema-os.xsd">
[a41]   <Subject>
[a42]   <Attribute
AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name">
[a43]    <AttributeValue>
[a44]    bs@simpsons.com
[a45]    </AttributeValue>
[a46]   </Attribute>
[a47]   </Subject>
[a48]   <Resource>
[a49]   <Attribute
AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#anyURI">
[a50]    <AttributeValue>
[a51]    file://example/med/record/patient/BartSimpson
[a52]    </AttributeValue>
[a53]   </Attribute>
[a54]   </Resource>
[a55]   <Action>
[a56]   <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-
id"
DataType="http://www.w3.org/2001/XMLSchema#string">
[a57]    <AttributeValue>
[a58]    read
[a59]    </AttributeValue>
[a60]   </Attribute>
[a61]   </Action>
[a62]   <Environment/>
[a63]  </Request>
```

В [a38] – [a40] содержится информация заголовка для контекста запроса и они используются также, заголовок для стратегии, разъяснения о котором были даны выше.

В элементе `<Subject>` содержится один или более атрибутов объекта, создающих запрос доступа. Может быть множество субъектов и у каждого субъекта может быть множество атрибутов. В этом случае в [a41] – [a47], имеется только один субъект и у этого субъекта есть только один атрибут: идентичность субъекта, выраженная, как имя электронной почты, "bs@simpsons.com". В этом примере атрибут `subject-category` пропущен. По этой причине принимается его значение по умолчанию – "access-subject".

В элементе `<Resource>` содержится один или более атрибутов ресурса, к которому субъект (или субъекты) запрашивают доступ. Может быть только один `<Resource>` на каждый запрос о принятии решения. В строках [a48] – [a54] содержится единственный атрибут ресурса, к которому запрашивал доступ Bart Simpson: ресурс, идентифицированный с помощью файла URI – "**file**://`medico/record/patient/BartSimpson`".

В элементе `<Action>` содержится один или более атрибутов действия, которое субъект (или субъекты) намерены совершить над ресурсом. Может быть только одно действие на каждый запрос о принятии решения. В [a55] – [a61] описана идентичность действия, которое Bart Simpson собирается предпринять, и это действие – "read".

Элемент `<Environment>`, [a62], является пустым.

В [a63] контекст запроса закрывается. В более сложном контексте запроса могут содержаться несколько атрибутов, не связанных с субъектом, ресурсом или действием. Они были бы помещены в необязательный элемент `<Environment>`, следующий за элементом `<Action>`.

Пункт PDP, обрабатывающий этот контекст запроса, помещает стратегию в свой репозиторий стратегий. Он сравнивает субъект, ресурс, действие и среду в контексте запроса с субъектами, ресурсами, действиями и средами в цели стратегии. Так как цель стратегии пуста, то стратегия сопоставляет этот контекст.

Теперь PDP сравнивает субъект, ресурс, действие и среду в контексте запроса с целью единственного правила в этой стратегии. Запрашиваемый ресурс сопоставим с элементом `<Target>` и запрашиваемое действие сопоставимо с элементом `<Target>`, но запрашиваемый атрибут "`subject`"-id не сопоставим с "med.example.com".

II.1.3 Пример контекста ответа

В результате оценки стратегии в этой стратегии отсутствует правило, которое возвращает результат "Permit" на этот запрос. Алгоритмом объединения правил для этой стратегии устанавливается, что в данном случае должен быть возвращен результат "NotApplicable". Контекст ответа выглядит следующим образом:

```
[a64]  <?xml version="1.0" encoding="UTF-8"?>
[a65]  <Response xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:context:schema:os
http://docs.oasis-open.org/xacml/xacml-core-2.0-context-schema-os.xsd">
[a66]   <Result>
[a67]   <Decision>NotApplicable</Decision>
[a68]   </Result>
[a69]  </Response>
```

В [a64] – [a65] содержится тот же самый вид информации заголовка для ответа, как и тот, что был описан выше для стратегии.

В элементе `<Result>` в строках [a66] – [a68] содержится результат оценки запроса о принятии решения относительно этой стратегии. В этом случае результатом будет "NotApplicable". Стратегия может возвращать "Permit", "Deny", "NotApplicable" или "Indeterminate". По этой причине требуется, чтобы PEP запретил доступ.

В [a69] контекст ответа закрывается.

II.2 Пример два

В этом пункте содержится пример документа XML, пример контекста запроса и пример правил XACML. Документом XML является медицинская карта. Определено четыре отдельных правила. Они иллюстрируют алгоритм объединения правил, условия и обязательства.

II.2.1 Пример экземпляра медицинской карты

Следующая запись является экземпляром медицинской карты, к которой может применяться пример правил XACML. Схема `<record>` определяется в зарегистрированном пространстве имен, которое управляется организацией Medi Corp.

```
[a70]  <?xml version="1.0" encoding="UTF-8"?>
[a71]   <record xmlns="urn:example:med:schemas:record"
[a72]   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
[a73]   <patient>
[a74]    <patientName>
[a75]    <first>Bartholomew</first>
[a76]    <last>Simpson</last>
[a77]    </patientName>
[a78]    <patientContact>
[a79]    <street>27 Shelbyville Road</street>
[a80]    <city>Springfield</city>
[a81]    <state>MA</state>
[a82]    <zip>12345</zip>
[a83]    <phone>555.123.4567</phone>
[a84]    <fax/>
[a85]    <email/>
[a86]    </patientContact>
[a87]    <patientDoB>1992-03-21</patientDoB>
[a88]    <patientGender>male</patientGender>
[a89]    <patient-number>555555</patient-number>
[a90]   </patient>
[a91]   <parentGuardian>
[a92]    <parentGuardianId>HS001</parentGuardianId>
[a93]    <parentGuardianName>
[a94]    <first>Homer</first>
[a95]    <last>Simpson</last>
[a96]    </parentGuardianName>
[a97]    <parentGuardianContact>
[a98]    <street>27 Shelbyville Road</street>
[a99]    <city>Springfield</city>
[a100]   <state>MA</state>
[a101]   <zip>12345</zip>
[a102]   <phone>555.123.4567</phone>
[a103]   <fax/>
[a104]   <email>homers@aol.com</email>
```

```
[a105]   </parentGuardianContact>
[a106]  </parentGuardian>
[a107]  <primaryCarePhysician>
[a108]   <physicianName>
[a109]   <first>Julius</first>
[a110]   <last>Hibbert</last>
[a111]   </physicianName>
[a112]   <physicianContact>
[a113]   <street>1 First St</street>
[a114]   <city>Springfield</city>
[a115]   <state>MA</state>
[a116]   <zip>12345</zip>
[a117]   <phone>555.123.9012</phone>
[a118]   <fax>555.123.9013</fax>
[a119]   <email/>
[a120]   </physicianContact>
[a121]   <registrationID>ABC123</registrationID>
[a122]   </primaryCarePhysician>
[a123]   <insurer>
[a124]   <name>Blue Cross</name>
[a125]   <street>1234 Main St</street>
[a126]   <city>Springfield</city>
[a127]   <state>MA</state>
[a128]   <zip>12345</zip>
[a129]   <phone>555.123.5678</phone>
[a130]   <fax>555.123.5679</fax>
[a131]   <email/>
[a132]   </insurer>
[a133]  <medical>
[a134]   <treatment>
[a135]   <drug>
[a136]    <name>methylphenidate hydrochloride</name>
[a137]    <dailyDosage>30mgs</dailyDosage>
[a138]    <startDate>1999-01-12</startDate>
[a139]   </drug>
[a140]   <comment>
[a141]    patient exhibits side-effects of skin coloration and carpal
degeneration
[a142]   </comment>
[a143]   </treatment>
[a144]   <result>
[a145]   <test>blood pressure</test>
[a146]   <value>120/80</value>
[a147]   <date>2001-06-09</date>
[a148]   <performedBy>Nurse Betty</performedBy>
[a149]   </result>
[a150]  </medical>
[a151]  </record>
```

II.2.2 Пример контекста запроса

В следующем примере иллюстрируется контекст запроса, к которому может быть применен пример правил. Он представляет запрос врача Julius Hibbert о считывании даты рождения пациента в карте Bartholomew Simpson.

```
[a152] <?xml version="1.0" encoding="UTF-8"?>
[a153] <Request xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=" urn:oasis:names:tc:xacml:2.0:context:schema:os
http://docs.oasis-open.org/xacml/access_control-xacml-2.0-context-schema-
os.xsd">
[a154]  <Subject>
[a155]  <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject-
category"
DataType="http://www.w3.org/2001/XMLSchema#anyURI">
[a156]   <AttributeValue>urn:oasis:names:tc:xacml:1.0:subject-
category:access-subject</AttributeValue>
[a157]  </Attribute>
[a158]  <Attribute
AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
 DataType="http://www.w3.org/2001/XMLSchema#string"
Issuer="med.example.com">
```

```
[a159]   <AttributeValue>CN=Julius Hibbert</AttributeValue>
[a160]  </Attribute>
[a161]  <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:name-
format"
DataType="http://www.w3.org/2001/XMLSchema#anyURI"
Issuer="med.example.com">
[a162]   <AttributeValue>
[a163]   urn:oasis:names:tc:xacml:1.0:datatype:x500name
[a164]   </AttributeValue>
[a165]  </Attribute>
[a166]  <Attribute
AttributeId="urn:oasis:names:tc:xacml:2.0:example:attribute:role"
DataType="http://www.w3.org/2001/XMLSchema#string"
Issuer="med.example.com">
[a167]   <AttributeValue>physician</AttributeValue>
[a168]  </Attribute>
[a169]  <Attribute
AttributeId="urn:oasis:names:tc:xacml:2.0:example:attribute:physician-id"
DataType="http://www.w3.org/2001/XMLSchema#string"
Issuer="med.example.com">
[a170]   <AttributeValue>jh1234</AttributeValue>
[a171]  </Attribute>
[a172]  </Subject>
[a173]  <Resource>
[a174]  <ResourceContent>
[a175]   <md:record xmlns:md="urn:example:med:schemas:record"
xsi:schemaLocation="urn:example:med:schemas:record
http:www.med.example.com/schemas/record.xsd">
[a176]   <md:patient>
[a177]    <md:patientDoB>1992-03-21</md:patientDoB>
[a178]    <md:patient-number>555555</md:patient-number>
[a179]   </md:patient>
[a180]   </md:record>
[a181]  </ResourceContent>
[a182]  <Attribute
AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
[a183]   <AttributeValue>
[a184]   //med.example.com/records/bart-simpson.xml#
[a185] xmlns(md=:Resource/ResourceContent/xpointer
[a186] (/md:record/md:patient/md:patientDoB)
[a187]   </AttributeValue>
[a188]  </Attribute>
[a189] </Resource>
[a190]  <Action>
[a191]  <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-
id"
DataType="http://www.w3.org/2001/XMLSchema#string">
[a192]   <AttributeValue>read</AttributeValue>
[a193]  </Attribute>
[a194]  </Action>
[a195]  <Environment/>
[a196] </Request>
```

[a152] – [a153] Стандартные описания пространства имен.

[a154] – [a172] Атрибуты субъекта помещаются в элемент `<Subject>` элемента `<Request>`. В каждом атрибуте содержатся атрибуты метаданных и значение атрибута. Есть только один субъект, участвующий в этом запросе.

[a155] – [a157] В каждом элементе `<Subject>` имеется атрибут `SubjectCategory`. Значение этого атрибута описывает роль, которую связанный субъект играет в создании запроса о принятии решения. Значение "`access-subject`" обозначает идентичность, для которой был выпущен этот запрос.

[a158] – [a160] Атрибут `subject-id` субъекта.

[a161] – [a165] Формат id-субъекта.

[a166] – [a168] Атрибут `role` субъекта.

[a169] – [a171] Атрибут `physician-id` субъекта.

[a173] – [a189] Атрибуты ресурса помещаются в элемент <Resource> элемента <Request>. Каждый атрибут состоит из метаданных атрибута и значения атрибута.

[a174] – [a181] Содержание ресурса. Экземпляр ресурса XML, к которому может запрашиваться доступ в полном объеме или частично, помещен сюда.

[a182] – [a188] Идентификатор экземпляра ресурса, к которому запрашивается доступ, являющийся выражением XPath внутри элемента <ResourceContent>; этот элемент выбирает данные, к которым нужен доступ.

[a190] – [a194] Атрибуты действия помещаются в элемент <Action> элемента <Request>.

[a192] Идентификатор действия.

[a195] Пустой элемент <Environment>.

II.2.3 Пример правил упрощенного языка

Должны осуществляться следующие правила упрощенного языка:

1) Правило 1: Личность, идентифицированная по своему номеру пациента, может считывать любые записи, по отношению к которым эта личность является обозначенным пациентом.
2) Правило 2: Личность может считывать любые записи, по отношению к которым эта личность является обозначенным родителем или опекуном и, если эти записи относятся к пациентам младше 16 лет.
3) Правило 3: Врач может делать записи в любом медицинском разделе, по отношению к которому этот врач является обозначенным основным лечащим врачом, при условии, что пациенту отправляется электронное сообщение.
4) Правило 4: Администратору не должно быть разрешено считывать или делать записи в медицинских разделах карты пациента.

Эти правила могут быть написаны разными пунктами PAP, действующими независимо, или единым PAP.

II.2.4 Пример экземпляра правил XACML

II.2.4.1 Правило 1

Правило 1 иллюстрирует простое правило с единым элементом <Condition>. Это также иллюстрирует использование элемента <VariableDefinition> для определения функции, которые могут использоваться посредством этой стратегии.

Следующий экземпляр <Rule> XACML выражает правило 1:

```
[a197] <?xml version="1.0" encoding="UTF-8"?>
[a198] <Policy
[a199] xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os"
[a200] xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=" urn:oasis:names:tc:xacml:2.0:policy:schema:os
http://docs.oasis-open.org/xacml/access_control-xacml-2.0-context-schema-
os.xsd"
[a201] xmlns:md="http://www.med.example.com/schemas/record.xsd"
[a202] PolicyId="urn:oasis:names:tc:xacml:2.0:example:policyid:1"
[a203] RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-
algorithm:deny-overrides">
[a204]  <PolicyDefaults>
[a205]        <XPathVersion>http://www.w3.org/TR/1999/Rec-xpath-
19991116</XPathVersion>
[a206]  </PolicyDefaults>
[a207]  <Target/>
[a208]  <VariableDefinition VariableId="17590034">
[a209]  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-
equal">
[a210]   <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-
one-and-only">
[a211]   <SubjectAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:2.0:example:attribute:patient-number"
[a212]   DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a213]   </Apply>
[a214]   <Apply
[a215]   FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-
only">
[a216]   <AttributeSelector
```

```
[a217]   RequestContextPath="//xacml-context:Resource/xacml-
context:ResourceContent/md:record/md:patient/md:patient-number/text()"
```

```
[a218]   DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a219]   </Apply>
[a220]  </Apply>
[a221]  </VariableDefinition>
[a222]  <Rule
[a223]  RuleId="urn:oasis:names:tc:xacml:2.0:example:ruleid:1"
[a224]  Effect="Permit">
[a225]  <Description>
[a226]   A person may read any medical record in the
[a227]   http://www.med.example.com/schemas/record.xsd namespace
[a228]   for which he or she is the designated patient
[a229]  </Description>
[a230]  <Target>
[a231]   <Resources>
[a232]   <Resource>
[a233]    <ResourceMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
[a234]    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">
[a235]     urn:example:med:schemas:record
[a236]    </AttributeValue>
[a237]    <ResourceAttributeDesignator AttributeId=
[a238]    "urn:oasis:names:tc:xacml:2.0:resource:target-namespace"
[a239]    DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a240]    </ResourceMatch>
[a241]    <ResourceMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:xpath-node-match">
[a242]    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">
[a243]     /md:record
[a244]    </AttributeValue>
[a245]    <ResourceAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:resource:xpath"
[a246]     DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a247]    </ResourceMatch>
[a248]   </Resource>
[a249]   </Resources>
[a250]   <Actions>
[a251]   <Action>
[a252]    <ActionMatch
[a253]    MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
[a254]    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">
[a255]      read
[a256]    </AttributeValue>
[a257]    <ActionAttributeDesignator
[a258]     AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
[a259]     DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a260]    </ActionMatch>
[a261]   </Action>
[a262]   </Actions>
[a263]  </Target>
[a264]  <Condition>
[a265]   <VariableReference VariableId="17590034"/>
[a266]  </Condition>
[a267]  </Rule>
[a268] </Policy>
```

[a199] – [a201] описания пространства имен XML.

[a205] Выражения XPath в этой стратегии должны интерпретироваться в соответствии с- W3C XPath:1999.

[a208] – [a221] Элемент `<VariableDefinition>`. Он определяет функцию, которая оценивает правдивость заявления: атрибут субъекта `patient-number` равен атрибуту `patient-number` в ресурсе.

[a209] Атрибут `FunctionId` именует функцию, которая должна использоваться для сравнения. В этом случае сравнение выполняется с функцией "`urn:oasis:names:tc:xacml:1.0:function:string-equal`"; эта функция принимает два аргумента типа "http://www.w3.org/2001/XMLSchema#string".

[a210] Первым аргументом определения переменной является функция, установленная атрибутом `FunctionId`. Так как `urn:oasis:names:tc:xacml:1.0:function:string-equal` принимает аргументы типа "http://www.w3.org/2001/XMLSchema#string" и `SubjectAttributeDesignator` выбирает "мешок" типа

"http://www.w3.org/2001/XMLSchema#string", то используется "urn:oasis:names:tc:xacml:1.0:function:string-one-and-only". Эта функция гарантирует, что ее аргумент оценивается как "мешок", в котором содержится только одно значение.

[a211] SubjectAttributeDesignator выбирает "мешок" значений для атрибута субъекта patient-number в контексте запроса.

[a215] Вторым аргументом определения переменной является функция, установленная атрибутом FunctionId. Так как "urn:oasis:names:tc:xacml:1.0:function:string-equal" принимает аргументы типа "http://www.w3.org/2001/XMLSchema#string" и AttributeSelector выбирает "мешок" типа "http://www.w3.org/2001/XMLSchema#string", то используется "urn:oasis:names:tc:xacml:1.0:function:string-one-and-only". Эта функция гарантирует, что ее аргумент оценивается как "мешок", в котором содержится только одно значение.

[a216] Элемент <AttributeSelector> выбирает "мешок" значений из контекста запроса, используя выражение XPath произвольной формы. В этом случае он выбирает значение patient-number в ресурсе. Заметим, что префиксы пространства имен в выражении XPath разрешены со стандартными описаниями пространства имен XML.

[a223] Идентификатор правила.

[a224] Описание эффекта правила. Если правило оценивается как 'True', то оно выдает значение атрибута Effect. Это значение затем объединяется со значениями Effect других правил, в соответствии с алгоритмом объединения правил.

[a225] – [a229] Описание правила в произвольной форме.

[a230] – [a263] Цель правила определяет набор запросов о принятии решения, который это правило будет оценивать. В этом примере пропущены элементы <Subjects> и <Environments>.

[a231] – [a249] В элементе <Resources> содержится дизъюнктивная последовательность элементов <Resource>. В этом примере имеется только одна последовательность.

[a232] – [a248] В элементе <Resource> заключена конъюнктивная последовательность элементов ResourceMatch. В этом примере имеется две последовательности.

[a233] – [a240] Первый элемент <ResourceMatch> сравнивает первый и второй дочерние элементы, в соответствии с функцией сопоставления. Сопоставление дает положительный результат, если значение первого аргумента сопоставимо с любыми значениями, выбранными вторым аргументом. Это сопоставление сравнивает пространство имен цели запрашиваемого документа со значением "urn:example:med:schemas:record".

[a233] Атрибут MatchId именует функцию сопоставления.

[a235] Буквенное значение атрибута для сопоставления.

[a237] – [a239] Элемент <ResourceAttributeDesignator> выбирает пространство имен цели из источника, содержащееся в контексте запроса. Имя атрибута устанавливается с помощью AttributeId.

[a241] – [a247] Второй элемент <ResourceMatch>. Это сопоставление сравнивает результаты двух выражений XPath. Вторым выражением XPath является расположение путей к запрашиваемому элементу XML и первым выражением XPath является буквенное значение "/md:record". Функция "xpath-node-match" оценивается как "True", если запрашиваемый элемент XML находится ниже элемента "/md:record".

[a250] – [a262] В элементе <Actions> содержится дизъюнктивная последовательность элементов <Action>. В этом случае имеется только один элемент <Action>.

[a251] – [a261] В элементе <Action> содержится конъюнктивная последовательность элементов <ActionMatch>. В этом случае имеется только один элемент <ActionMatch>.

[a252] – [a260] Элемент <ActionMatch> сравнивает свои первый и второй дочерние элементы в соответствии с функцией сопоставления. Сопоставление считается положительным, если значение первого аргумента сопоставимо с любыми значениями, выбранными вторым аргументом. В этом случае значение атрибута действия action-id в контексте запроса сравнивается с буквенным значением "read".

[a264] – [a266] Элемент <Condition>. Условие должно оцениваться, как "True" для того правила, которое применимо. В этом условии содержится ссылка на определение переменной, определенной повсюду в этой стратегии.

II.2.4.2 Правило 2

Правило 2 иллюстрирует использование математической функции, т. е. элемента <Apply> с FunctionId "urn:oasis:names:tc:xacml:1.0:function:date-add-yearMonthDuration" для вычисления даты шестнадцатилетия пациента. Оно также иллюстрирует использование выражений предикат с FunctionId "urn:oasis:names:tc:xacml:1.0:function:and". В этом примере имеется одна функция, встроенная в элемент <Condition> и еще одна функция, на которую ссылаются в элементе <VariableDefinition>.

```
[a269] <?xml version="1.0" encoding="UTF-8"?>
[a270] <Policy
[a271] xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os"
[a272] xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-os.xsd"
[a273] xmlns:xf="urn:oasis:names:tc:xacml:2.0:data-types"
[a274] xmlns:md="http:www.med.example.com/schemas/record.xsd"
[a275] PolicyId="urn:oasis:names:tc:xacml:2.0:example:policyid:2"
RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-
overrides">
[a276]  <PolicyDefaults>
[a277]  <XPathVersion>http://www.w3.org/TR/1999/Rec-xpath-19991116</XPathVersion>
[a278]  </PolicyDefaults>
[a279]  <Target/>
[a280]  <VariableDefinition VariableId="17590035">
[a281]  <Apply FunctionId="urn:oasis:names:tc:xacml:2.0:function:date-less-or-
equal">
[a282]   <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-one-and-
only">
[a283]   <EnvironmentAttributeDesignator
[a284]   AttributeId= "urn:oasis:names:tc:xacml:1.0:environment:current-date"
[a285]   DataType="http://www.w3.org/2001/XMLSchema#date"/>
[a286]   </Apply>
[a287]   <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-add-
yearMonthDuration">
[a288]   <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-one-and-
only">
[a289]    <AttributeSelector RequestContextPath=
[a290]    "//md:record/md:patient/md:patientDoB/text()"
[a291]    DataType="http://www.w3.org/2001/XMLSchema#date"/>
[a292]   </Apply>
[a293]   <AttributeValue
[a294]   DataType="urn:oasis:names:tc:xacml:2.0:data-types:yearMonthDuration">
[a295]    <xf:dt-yearMonthDuration>
[a296]    P16Y
[a297]    </xf:dt-yearMonthDuration>
[a298]   </AttributeValue>
[a299]   </Apply>
[a300]  </Apply>
[a301]  </VariableDefinition>
[a302]  <Rule
[a303]  RuleId="urn:oasis:names:tc:xacml:2.0:example:ruleid:2"
[a304]  Effect="Permit">
[a305]  <Description>
[a306]   A person may read any medical record in the
[a307]   http://www.med.example.com/records.xsd namespace
[a308]   for which he or she is the designated parent or guardian,
[a309]   and for which the patient is under 16 years of age
[a310]  </Description>
[a311]  <Target>
[a312]   <Resources>
[a313]   <Resource>
[a314]    <ResourceMatch
[a315]    MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
[a316]    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
[a317]      http://www.med.example.com/schemas/record.xsd
[a318]     </AttributeValue>
[a319]    <ResourceAttributeDesignator AttributeId=
"urn:oasis:names:tc:xacml:2.0:resource:target-namespace"
[a320]    DataType="http://www.w3.org/2001/XMLSchema#string"/>
```

```
[a321]    </ResourceMatch>
[a322]    <ResourceMatch
[a323]    MatchId="urn:oasis:names:tc:xacml:1.0:function:xpath-node-match">
[a324]    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
[a325]     /md:record
[a326]    </AttributeValue>
[a327]    <ResourceAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:resource:xpath"
[a328]    DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a329]    </ResourceMatch>
[a330]   </Resource>
[a331]   </Resources>
[a332]   <Actions>
[a333]   <Action>
[a334]    <ActionMatch
[a335]    MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
[a336]    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
[a337]     read
[a338]    </AttributeValue>
[a339]    <ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
[a340]    DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a341]   </ActionMatch>
[a342]   </Action>
[a343]   </Actions>
[a344]  </Target>
[a345]  <Condition>
[a346]  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
[a347]   <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
[a348]   <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-
only">
[a349]    <SubjectAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:2.0:example:attribute:
[a350] parent-guardian-id"
[a351]    DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a352]   </Apply>
[a353]   <Apply
[a354]   FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
[a355]    <AttributeSelector
[a356]    RequestContextPath="//xacml-context:Resource/xacml-
context:ResourceContent/md:record/md:parentGuardian/md:parentGuardianId/text()"
[a357]    DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a358]    </Apply>
[a359]   </Apply>
[a360]   <VariableReference VariableId="17590035"/>
[a361]   </Apply>
[a362]  </Condition>
[a363]  </Rule>
[a364] </Policy>
```

[a280] – [a301] В элементе `<VariableDefinition>` содержится часть условия (т. е. пациенту еще не исполнилось 16 лет?). Пациенту нет 16 лет, если текущая дата меньше, чем дата, вычисленная добавлением 16 к дате рождения пациента.

[a281] – [a300] "`urn:oasis:names:tc:xacml:1.0:function:date-less-or-equal`" используется для вычисления разности аргументов даты.

[a282] – [a286] В первом аргументе даты используется "`urn:oasis:names:tc:xacml:1.0:function:date-one-and-only`" для гарантии того, что "мешок" значений, выбранный ее аргументом, содержит точно одно значение типа "http://www.w3.org/2001/XMLSchema#date".

[a284] Текущая дата оценивается выбором атрибута среды "`urn:oasis:names:tc:xacml:1.0:environment:current-date`".

[a287] – [a299] Второй аргумент даты использует "urn:oasis:names:tc:xacml:1.0:function:date-add-yearMonthDuration" для вычисления даты шестнадцатилетия пациента, добавляя 16 лет к дате рождения пациента. У первого из ее аргументов тип "http://www.w3.org/2001/XMLSchema#date", а у второго – тип "urn:oasis:names:tc:xacml:2.0:data-types:yearMonthDuration".

[a289] Элемент <AttributeSelector> выбирает дату рождения пациента, принимая выражение XPath в содержании ресурса.

[a293] – [a298] Длительность Year Month Duration – 16 лет.

[a311] – [a344] Описание правила и цель правила. Смотрите Правило 1 в пункте II.4.2.4.1 для более подробного объяснения для этих элементов.

[a345] – [a362] Элемент <Condition>. Условие должно оцениваться в "True" для того правила, которое применимо. Это условие оценивает правдивость заявления: запросчиком является обозначенный родитель или опекун и возраст пациента менее 16 лет. В нем содержится один встроенный элемент <Apply> и один упомянутый элемент <VariableDefinition>.

[a346] Условие использует функцию "urn:oasis:names:tc:xacml:1.0:function:and". Это логическая функция, которая принимает один или более логических аргументов (в этом случае 2) и выполняет операцию логического "AND" для вычисления настоящего значения выражения.

[a347] – [a359] Оценивается первая часть этого условия (т. е. является ли запросчиком родитель или опекун?). Функцией является "urn:oasis:names:tc:xacml:1.0:function:string-equal" и она принимает два аргумента типа "http://www.w3.org/2001/XMLSchema#string".

В [a348] назначается первый аргумент. Так как "urn:oasis:names:tc:xacml:1.0:function:string-equal" принимает аргументы типа "http://www.w3.org/2001/XMLSchema#string", "urn:oasis:names:tc:xacml:1.0:function:string-one-and-only" используется для гарантирования того, что в атрибуте субъекта "urn:oasis:names:tc:xacml:2.0:example:attribute:parent-guardian-id" в контексте запроса содержится точно одно значение.

В [a353] назначается второй аргумент. Значение атрибута субъекта "urn:oasis:names:tc:xacml:2.0:example:attribute:parent-guardian-id" выбирается из контекста запроса, с помощью использования элемента <SubjectAttributeDesignator>.

[a354] Как было указано выше, "urn:oasis:names:tc:xacml:1.0:function:string-one-and-only" используется для гарантирования того, что в "мешке" значений, выбранном ее аргументом, содержится точно одно значение типа "http://www.w3.org/2001/XMLSchema#string".

[a355] Второй аргумент выбирает значение элемента <md:parentGuardianId> из содержания ресурса, используя элемент <AttributeSelector>. В этом элементе содержится выражение XPath в произвольной форме, указывающее на контекст запроса. Заметим, что все префиксы пространства имен в выражении XPath разрешены со стандартными описаниями пространства имен. AttributeSelector оценивается как "мешок" значений типа "http://www.w3.org/2001/XMLSchema#string".

[a360] ссылается на элемент <VariableDefinition>, в котором определяется вторая часть условия.

II.2.4.3 Правило 3

Правило 3 иллюстрирует использование обязательства. В синтаксис элемента <Rule> XACML <Rule> не включен элемент, подходящий для переноса обязательства, вот почему Правило 3 должно быть форматировано, как элемент <Policy>.

```
[a365] <?xml version="1.0" encoding="UTF-8"?>
[a366] <Policy
[a367]  xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os"
[a368]  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
[a369]  xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-
os.xsd"
[a370] xmlns:md="http:www.med.example.com/schemas/record.xsd"
[a371]  PolicyId="urn:oasis:names:tc:xacml:2.0:example:policyid:3"
[a372]  RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-
algorithm:deny-overrides">
[a373]  <Description>
[a374]  Policy for any medical record in the
[a375]  http://www.med.example.com/schemas/record.xsd namespace
[a376]  </Description>
[a377]  <PolicyDefaults>
```

```
[a378]  <XPathVersion>http://www.w3.org/TR/1999/Rec-xpath-
19991116</XPathVersion>
[a379]  </PolicyDefaults>
[a380]  <Target>
[a381]  <Resources>
[a382]   <Resource>
[a383]   <ResourceMatch
[a384]   MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
[a385]    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">
[a386]    urn:example:med:schemas:record
[a387]    </AttributeValue>
[a388]    <ResourceAttributeDesignator AttributeId=
[a389]    "urn:oasis:names:tc:xacml:1.0:resource:target-namespace"
[a390]    DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a391]   </ResourceMatch>
[a392]   </Resource>
[a393]  </Resources>
[a394]  </Target>
[a395]  <Rule RuleId="urn:oasis:names:tc:xacml:2.0:example:ruleid:3"
[a396]  Effect="Permit">
[a397]  <Description>
[a398]   A physician may write any medical element in a record
[a399]   for which he or she is the designated primary care
[a400]   physician, provided an email is sent to the patient
[a401]  </Description>
[a402]  <Target>
[a403]   <Subjects>
[a404]   <Subject>
[a405]    <SubjectMatch
[a406]    MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
[a407]    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">
[a408]     physician
[a409]     </AttributeValue>
[a410]     <SubjectAttributeDesignator AttributeId=
"urn:oasis:names:tc:xacml:2.0:example:attribute:role"
[a411]     DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a412]    </SubjectMatch>
[a413]   </Subject>
[a414]   </Subjects>
[a415]   <Resources>
[a416]   <Resource>
[a417]    <ResourceMatch
[a418]    MatchId="urn:oasis:names:tc:xacml:1.0:function:xpath-node-match">
[a419]    <AttributeValue
[a420]     DataType="http://www.w3.org/2001/XMLSchema#string">
[a421]      /md:record/md:medical
[a422]     </AttributeValue>
[a423]    <ResourceAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:resource:xpath"
[a424]    DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a425]    </ResourceMatch>
[a426]   </Resource>
[a427]   </Resources>
[a428]   <Actions>
[a429]   <Action>
[a430]    <ActionMatch
[a431]    MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
[a432]    <AttributeValue
[a433]    DataType="http://www.w3.org/2001/XMLSchema#string">
[a434]     write
[a435]    </AttributeValue>
[a436]    <ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
[a437]    DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a438]    </ActionMatch>
[a439]   </Action>
[a440]   </Actions>
[a441]  </Target>
```

```
[a442]  <Condition>
[a443]  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-
equal">
[a444]   <Apply
[a445]   FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-
only">
[a446]   <SubjectAttributeDesignator
[a447]   AttributeId="urn:oasis:names:tc:xacml:2.0:example:
attribute:physician-id"
[a448]   DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a449]   </Apply>
[a450]   <Apply
[a451]   FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-
only">
[a452]   <AttributeSelector RequestContextPath=
[a453]   "//xacml-context:Resource/xacml-
context:ResourceContent/md:record/md:primaryCarePhysician/md:registrationID
/text()"
[a454]   DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a455]   </Apply>
[a456]   </Apply>
[a457]  </Condition>
[a458]  </Rule>
[a459]  <Obligations>
[a460]  <Obligation
ObligationId="urn:oasis:names:tc:xacml:example:obligation:email"
[a461]  FulfillOn="Permit">
[a462]   <AttributeAssignment
AttributeId="urn:oasis:names:tc:xacml:2.0:example:attribute:mailto"
[a463]   DataType="http://www.w3.org/2001/XMLSchema#string">
[a464]   &lt;AttributeSelector RequestContextPath=
[a465]   "//md:/record/md:patient/md:patientContact/md:email"
[a466]   DataType="http://www.w3.org/2001/XMLSchema#string"/&gt ;
[a467]   </AttributeAssignment>
[a468]   <AttributeAssignment
AttributeId="urn:oasis:names:tc:xacml:2.0:example:attribute:text"
[a469]   DataType="http://www.w3.org/2001/XMLSchema#string">
[a470]    Your medical record has been accessed by:
[a471]   </AttributeAssignment>
[a472]   <AttributeAssignment
AttributeId="urn:oasis:names:tc:xacml:2.0:example:attribute:text"
[a473]   DataType="http://www.w3.org/2001/XMLSchema#string">
[a474]   &lt;SubjectAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
[a475]   DataType="http://www.w3.org/2001/XMLSchema#string"/&gt;
[a476]   </AttributeAssignment>
[a477]  </Obligation>
[a478]  </Obligations>
[a479] </Policy>
```

[a366] – [a372] В элемент `<Policy>` включены стандартные описания пространства имен, а также конкретные параметры стратегии, такие как `PolicyId` и `RuleCombiningAlgId`.

[a371] Идентификатор стратегии. Этот параметр разрешает ссылку на стратегию с помощью стратегического набора.

[a372] Алгоритм объединения правил идентифицирует алгоритм для объединения результатов оценки правил.

[a373] – [a376] Описание этой стратегии в произвольной форме.

[a379] – [a394] Цель стратегии. В цели стратегии определяется набор применимых запросов о принятии решения. Структура элемента `<Target>` в `<Policy>` идентична структуре элемента `<Target>` в `<Rule>`. В этом случае целью стратегии является набор всех ресурсов XML, которые совместимы с пространством имен "`urn:example:med:schemas:record`".

[a395] Единственный элемент `<Rule>` включен в этот элемент `<Policy>`. В заголовке правила установлено два параметра: `RuleId` and `Effect`.

[a402] – [a441] Цель правила в еще большей степени ограничивает эту цель стратегии.

[a405] – [a412] Элемент `<SubjectMatch>` нацеливает правило на субъекты, чей атрибут субъекта "`urn:oasis:names:tc:xacml:2.0:example:attribute:role`" равен значению "`physician`".

[a417] – [a425] Элемент <ResourceMatch> нацеливает правило на ресурсы, которые сопоставимы с выражением XPath "/md:record/md:medical".

[a430] – [a438] Элемент <ActionMatch> нацеливает правило на действия, атрибут действия которых "urn:oasis:names:tc:xacml:1.0:action:action-id" равен значению "write".

[a442] – [a457] Элемент <Condition>. Для того чтобы правило было применимо к запросу о принятии решения, условие должно оцениваться, как "True". В этом условии сравнивается значение атрибута субъекта "urn:oasis:names:tc:xacml:2.0:example:attribute:physician-id" со значением элемента <registrationId> в медицинской карте, к которому нужен доступ.

[a459] – [a478] Элемент <Obligations>. Обязательствами является набор действий, которые должны быть выполнены с помощью PEP вместе с решением об авторизации. Обязательство может быть связано с решением об авторизации "Permit" или "Deny". В элементе содержится единое обязательство.

[a460] – [a477] Элемент <Obligation> состоит из атрибута ObligationId, значении решения об авторизации, для которого он должен быть выполнен и набора присвоенных значений атрибута. PDP не принимает решение о присвоенных значениях атрибута. Это работа пункта PEP.

[a460] Атрибут ObligationId идентифицирует обязательство. В этом случае пунктом PEP требуется отправление электронного сообщения.

[a461] В атрибуте FulfillOn определяется значение решения об авторизации, для которого должно быть выполнено это обязательство. В этом случае, если разрешен доступ.

[a462] – [a467] Первый параметр показывает, где в ресурсе будет найден электронный адрес пунктом PEP.

[a468] – [a471] Во втором параметре содержится буквенный текст для основной части электронного сообщения.

[a472] – [a476] Третий параметр показывает, где в ресурсе будет найден дальнейший текст для основной части электронного сообщения.

II.2.4.4 Правило 4

Правило 4 иллюстрирует использование значения "Deny" Effect и <Rule> без элемента <Condition>.

```
[a480] <?xml version="1.0" encoding="UTF-8"?>
[a481] <Policy
[a482] xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
[a483] xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-
os.xsd"
[a484] xmlns:md="http:www.med.example.com/schemas/record.xsd"
[a485] PolicyId="urn:oasis:names:tc:xacml:2.0:example:policyid:4"
[a486] RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-
algorithm:deny-overrides">
[a487]  <PolicyDefaults>
[a488]  <XPathVersion>http://www.w3.org/TR/1999/Rec-xpath-
19991116</XPathVersion>
[a489]  </PolicyDefaults>
[a490]  <Target/>
[a491]  <Rule
[a492]  RuleId="urn:oasis:names:tc:xacml:2.0:example:ruleid:4"
[a493]  Effect="Deny">
[a494]  <Description>
[a495]   An Administrator shall not be permitted to read or write
[a496]   medical elements of a patient record in the
[a497]   http://www.med.example.com/records.xsd namespace.
[a498]  </Description>
[a499]  <Target>
[a500]   <Subjects>
[a501]   <Subject>
[a502]    <SubjectMatch
[a503]    MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
[a504]    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">
[a505]     administrator
[a506]    </AttributeValue>
[a507]    <SubjectAttributeDesignator AttributeId=
[a508]    "urn:oasis:names:tc:xacml:2.0:example:attribute:role"
[a509]    DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a510]    </SubjectMatch>
```

```
[a511]   </Subject>
[a512]   </Subjects>
[a513]   <Resources>
[a514]   <Resource>
[a515]    <ResourceMatch
[a516]    MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
[a517]    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">
[a518]     urn:example:med:schemas:record
[a519]    </AttributeValue>
[a520]    <ResourceAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:resource:target-namespace"
[a521]    DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a522]    </ResourceMatch>
[a523]    <ResourceMatch
[a524]    MatchId="urn:oasis:names:tc:xacml:1.0:function:xpath-node-match">
[a525]    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">
[a526]     /md:record/md:medical
[a527]    </AttributeValue>
[a528]    <ResourceAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:resource:xpath"
[a529]    DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a530]    </ResourceMatch>
[a531]   </Resource>
[a532]   </Resources>
[a533]   <Actions>
[a534]   <Action>
[a535]    <ActionMatch
[a536]    MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
[a537]    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">
[a538]     read
[a539]    </AttributeValue>
[a540]    <ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
[a541]    DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a542]    </ActionMatch>
[a543]   </Action>
[a544]   <Action>
[a545]    <ActionMatch
[a546]    MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
[a547]    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">
[a548]     write
[a549]    </AttributeValue>
[a550]    <ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
[a551]    DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a552]    </ActionMatch>
[a553]   </Action>
[a554]   </Actions>
[a555]  </Target>
[a556]  </Rule>
[a557] </Policy>
```

[a492] – [a493] The `<Rule>` element declaration.

[a493] Правило `Effect`. Каждое правило, которое оценивается как "True" выдает эффект правила, как свое значение. Значением `Effect` этого правила является "Deny", а в соответствии с этим правилом, доступ должен быть запрещен, если оно оценивается как "True".

[a494] – [a498] Описание этого правила в свободной форме.

[a499] – [a555] Цель правила. Цель правила определяет набор запросов о принятии решения, которые применимы к этому правилу.

[a502] – [a510] Элемент `<SubjectMatch>` нацеливает это правило на субъекты, чей атрибут субъекта "`urn:oasis:names:tc:xacml:2.0:example:attribute:role`" равен значению "administrator".

[a513] – [a532] В элементе <Resources> содержится один элемент <Resource>, который (в свою очередь) содержит два элемента <ResourceMatch>. Цель сопоставима, если ресурс, идентифицированный контекстом запроса, сопоставим с обоими критериями сопоставления ресурса.

[a515] – [a522] Первый элемент <ResourceMatch> нацеливает правило на ресурсы, чей атрибут ресурса "urn:oasis:names:tc:xacml:2.0:resource:target-namespace" равен "urn:example:med:schemas:record".

[a523] – [a530] Второй элемент <ResourceMatch> нацеливает это правило на элементы XML, которые сопоставимы с выражением XPath "/md:record/md:medical".

[a533] – [a554] В элементе <Actions> содержится два элемента <Action>, в каждом из которых содержится один элемент <ActionMatch>. Цель сопоставима, если действие, идентифицированное в контексте запроса, сопоставимо с любым из двух критериев сопоставления действия.

[a535] – [a552] Элементы <ActionMatch> нацеливают это правило на действия, чей атрибут действия "urn:oasis:names:tc:xacml:1.0:action:action-id" равен значению "read" or "write".

В этом правиле нет элемента <Condition>.

II.2.4.5 Пример PolicySet

В этом пункте используются примеры предыдущих пунктов для иллюстрации процесса объединения стратегий. Стратегия, управляющая доступом к считыванию медицинских разделов в карте, формируется из каждого из этих четырех правил, описанных в пункте II.4.2.3. На упрощенном языке объединенным правилом является:

- Или запросчик является пациентом; или
- Запросчик является родителем или опекуном и пациенту меньше 16 лет; или
- Запросчик является основным лечащим врачом, и уведомление высылается пациенту врачом; и
- Запросчик не является администратором.

Следующий набор стратегий иллюстрирует объединенные стратегии. Стратегия 3 включается с помощью ссылки, а Стратегия 2 включена явным образом.

```
[a558] <?xml version="1.0" encoding="UTF-8"?>
[a559]  <PolicySet
[a560]  xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
[a561]  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-
os.xsd"
[a562]  PolicySetId=
[a563]  "urn:oasis:names:tc:xacml:2.0:example:policysetid:1"
[a564]  PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:
[a565]  policy-combining-algorithm:deny-overrides">
[a566]  <Description>
[a567]   Example policy set.
[a568]  </Description>
[a569]  <Target>
[a570]   <Resources>
[a571]   <Resource>
[a572]    <ResourceMatch
[a573]    MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
[a574]    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">
[a575]     urn:example:med:schema:records
[a576]    </AttributeValue>
[a577]    <ResourceAttributeDesignator AttributeId=
[a578]    "urn:oasis:names:tc:xacml:1.0:resource:target-namespace"
[a579]    DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a580]    </ResourceMatch>
[a581]   </Resource>
[a582]   </Resources>
[a583]  </Target>
[a584]  <PolicyIdReference>
[a585]  urn:oasis:names:tc:xacml:2.0:example:policyid:3
[a586]  </PolicyIdReference>
[a587]  <Policy
[a588]  PolicyId="urn:oasis:names:tc:xacml:2.0:example:policyid:2"
```

```
[a589]  RuleCombiningAlgId=
[a590] "urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-
overrides">
[a591]  <Target/>
[a592]  <Rule RuleId="urn:oasis:names:tc:xacml:2.0:example:ruleid:1"
[a593]  Effect="Permit">
[a594]  </Rule>
[a595]  <Rule RuleId="urn:oasis:names:tc:xacml:2.0:example:ruleid:2"
[a596]  Effect="Permit">
[a597]  </Rule>
[a598]  <Rule RuleId="urn:oasis:names:tc:xacml:2.0:example:ruleid:4"
[a599]  Effect="Deny">
[a600]  </Rule>
[a601]  </Policy>
[a602] </PolicySet>
```

[a559] – [a565] Описание элемента `<PolicySet>`. Включены стандартные описания пространства имен XML.

[a562] Используется атрибут `PolicySetId` для идентификации этого стратегического набора для возможного его включения в другой стратегический набор.

[a564] Идентификатор алгоритма объединения стратегий. Стратегии и стратегические наборы в этом стратегическом наборе объединены в соответствии с установленным алгоритмом объединения стратегий, если вычислено решение об авторизации.

[a566] – [a568] Описание стратегического набора в произвольной форме.

[a569] – [a583] В элементе `<Target>` стратегического набора определяется набор запросов о принятии решения, применимый к этому элементу `<PolicySet>`.

[a584] В `PolicyIdReference` включается стратегия с помощью id.

[a588] `Policy 2` явно включается в этот стратегический набор. Правила в `Policy 2` пропущены для понятности.

Дополнение III

Описание примера функций "мешка" более высокого порядка

III.1 Пример функций "мешка" более высокого порядка

В этом дополнении описаны функции в XACML, которые выполняют действия с "мешками" таким образом, чтобы функции можно было применять к "мешкам" вообще.

Для целей примера, функциональный язык общего назначения, называемый Haskell (смотрите [Haskell]) используется для установки семантики этих функций. Хотя описания на английском языке достаточно, все же полезна и надлежащая спецификация этой семантики.

Для быстрого обзора, в последующей системе обозначений Haskell, определение функции принимает форму пунктов, которые применяются к образцам структур, а именно, спискам. Символ "[]" обозначает пустой список, в то время, как выражение "(x:xs)" сопоставляется с аргументом непустого списка, в котором "x" представляет первый элемент этого списка, а "xs" является оставшейся частью этого списка. Мы используем систему обозначений Haskell списка, которая является упорядоченной совокупностью элементов, для моделирования "мешков" значений XACML.

Простое определение Haskell знакомой функции "urn:oasis:names:tc:xacml:1.0:function:and", которая принимает список значений логического типа, определяется следующим образом:

```
and:: [Bool]     -> Bool
and []           = True
and (x:xs)       = x && (and xs)
```

Первая строка определения, обозначенная с помощью "::", формально описывает тип данных функции, которая принимает список логических значений, обозначенных с помощью "[Bool]", и возвращает логическое значение, обозначенное с помощью "Bool". Вторая строка определения является пунктом, который заявляет, что функцией "and", примененной к пустому списку, является "True". Третьей строкой определения является пункт, который заявляет, что для непустого списка, такого, в котором первый элемент "x", являющийся значением логического типа данных, функция "and", примененная к x, должна быть объединена с, используя логическую функцию конъюнкции, обозначенную инфиксным символом "&&", результатом рекурсивно примененной функции "and" к оставшейся части списка. Конечно, применение функции "and" является "True", если и только если список, к которому она применяется, пустой или каждый элемент этого списка "True". Например, оценка следующих выражений Haskell,

(and []), (and [True]), (and [True,True]), (and [True,True,False])

Оценивается как "True", "True", "True", и "False", соответственно.

1) `urn:oasis:names:tc:xacml:1.0:function:any-of`

В Haskell, семантика этой операции следующая:

any_of :: (a -> b -> Bool) -> a -> [b] -> Bool

any_of f a [] = False

any_of f a (x:xs) = (f a x) || (any_of f a xs)

В вышеуказанной системе обзначений, "f" является функцией, которую нужно применить, "a" является простым значением, а "(x:xs)" представляет первый элемент списка, как "x", а оставшийся список, как "xs".

Например, следующее выражение должно возвратить "True":

```
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
       <Function
              FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-   equal"/>
       <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
              Paul
       </AttributeValue>
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
                     John
              </AttributeValue>
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
                     Paul
              </AttributeValue>
              <AttributeValue
                     DataType="http://www.w3.org/2001/XMLSchema#string">George
              </AttributeValue>
              <AttributeValue
                     DataType="http://www.w3.org/2001/XMLSchema#string">
                     Ringo
              </AttributeValue>
       </Apply>
</Apply>
```

Это выражение – "True" потому, что первый аргумент равен, по крайней мере, одному из элементов этого "мешка", в соответствии с функцией.

2) `urn:oasis:names:tc:xacml:1.0:function:all-of`

В Haskell, семантика этой операции следующая:

all_of :: (a -> b -> Bool) -> a -> [b] -> Bool

all_of f a [] = True

all_of f a (x:xs) = (f a x) && (all_of f a xs)

В вышеуказанной системе обзначений, "f" является функцией, которую нужно применить, "a" является простым значением, а "(x:xs)" представляет первый элемент списка, как "x", а оставшийся список, как "xs".

Например, следующее выражение должно быть оценено, как "True":

```
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:all-of">
       <Function FunctionId="urn:oasis:names:tc:xacml:2.0:function:integer-
       greater"/>
       <AttributeValue
       DataType="http://www.w3.org/2001/XMLSchema#integer">10</AttributeValue>
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-bag">
              <AttributeValue
       DataType="http://www.w3.org/2001/XMLSchema#integer">9</AttributeValue>
              <AttributeValue
       DataType="http://www.w3.org/2001/XMLSchema#integer">3</AttributeValue>
              <AttributeValue
       DataType="http://www.w3.org/2001/XMLSchema#integer">4</AttributeValue>
              <AttributeValue
       DataType="http://www.w3.org/2001/XMLSchema#integer">2</AttributeValue>
       </Apply>
</Apply>
```

Это выражение – "True" потому, что первый аргумент (10) больше, чем все элементы этого "мешка" (9,3,4 и 2).

3) `urn:oasis:names:tc:xacml:1.0:function:any-of-any`

В Haskell, пользуясь функцией "any_of", определенной выше, семантика функции "any_of_any" будет следующей:

```
any_of_any :: ( a -> b -> Bool )    -> [a]-> [b] -> Bool
any_of_any f []                     ys          = False
any_of_any f (x:xs)                 ys          = (any_of f x ys) || (any_of_any f xs ys)
```

В вышеуказанной системе обозначений, "f" является функцией, которую нужно применить, а "(x:xs)" представляет первый элемент списка, как "x", а оставшийся список, как "xs".

Например, следующее выражение должно быть оценено, как "True":

```
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">
       <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-
equal"/>
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
              <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Ringo</AttributeValue>
              <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Mary</AttributeValue>
       </Apply>
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
              <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">John</AttributeValue>
              <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Paul</AttributeValue>
              <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">George</AttributeValue>
              <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Ringo</AttributeValue>
       </Apply>
</Apply>
```

Это выражение – "True" потому, что. по крайней мере, один из элементов первого "мешка", а именно, "Ringo", равен, по крайней мере, одному из элементов второго "мешка".

4) `urn:oasis:names:tc:xacml:1.0:function:all-of-any`

В Haskell, пользуясь функцией "any_of", определенной выше, семантика функции "any_of_any" будет следующей:

```
all_of_any :: ( a -> b -> Bool )    -> [a]-> [b] -> Bool
all_of_any  f []            ys              = True
all_of_any  f (x:xs)        ys              = (any_of f x ys) && (all_of_any f xs ys)
```

В вышеуказанной системе обозначений, "f" является функцией, которую нужно применить, а "(x:xs)" представляет первый элемент списка, как "x", а оставшийся список, как "xs".

Например, следующее выражение должно быть оценено, как "True":

```
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:all-of-any">
       <Function FunctionId="urn:oasis:names:tc:xacml:2.0:function:integer-
greater"/>
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-bag">
              <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">10</AttributeValue>
              <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">20</AttributeValue>
       </Apply>
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-bag">
              <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">1</AttributeValue>
```

```
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">3</AttributeValue>
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">5</AttributeValue>
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">19</AttributeValue>
      </Apply>
</Apply>
```

Это выражение – "True" потому, что каждый из элементов первого мешка больше чем, по крайней мере, один из элементов второго мешка.

5) `urn:oasis:names:tc:xacml:1.0:function:any-of-all`

В Haskell, пользуясь функцией "any_of", определенной выше, семантика функции "any_of_any" будет следующей:

```
any_of_all :: ( a -> b -> Bool )        -> [a]-> [b] -> Bool
any_of_all f []    ys                   = False
any_of_all f (x:xs)        ys                   = (all_of f x ys) || ( any_of_all f xs ys)
```

В вышеуказанной системе обозначений, "f" является функцией, которую нужно применить, а "(x:xs)" представляет первый элемент списка, как "x", а оставшийся список, как "xs".

Например, следующее выражение должно быть оценено, как "True":

```
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-all">
      <Function FunctionId="urn:oasis:names:tc:xacml:2.0:function:integer-
greater"/>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-bag">
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">3</AttributeValue>
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">5</AttributeValue>
      </Apply>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-bag">
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">1</AttributeValue>
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">2</AttributeValue>
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">3</AttributeValue>
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">4</AttributeValue>
      </Apply>
</Apply>
```

Это выражение – "True" потому, что для всех значений во втором "мешке" имеется большее значение в первом "мешке".

6) `urn:oasis:names:tc:xacml:1.0:function:all-of-all`

В Haskell, пользуясь функцией "any_of", определенной выше, семантика функции "any_of_any" будет следующей:

```
all_of_all :: ( a -> b -> Bool )       -> [a] -> [b] -> Bool
all_of_all  f []    ys                 = True
all_of_all  f (x:xs)        ys                 = (all_of f x ys) && (all_of_all f xs ys)
```

В вышеуказанной системе обозначений, "f" является функцией, которую нужно применить, а "(x:xs)" представляет первый элемент списка, как "x", а оставшийся список, как "xs".

Например, следующее выражение должно быть оценено, как "True":

```
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:all-of-all">
      <Function FunctionId="urn:oasis:names:tc:xacml:2.0:function:integer-
greater"/>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-bag">
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">6</AttributeValue>
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">5</AttributeValue>
      </Apply>
```

```
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-bag">
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">1</AttributeValue>
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">2</AttributeValue>
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">3</AttributeValue>
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">4</AttributeValue>
      </Apply>
</Apply>
```

Это выражение – "True" потому, что все элементы первого мешка, "5" и "6", каждое из которых больше, чем все целочисленные значения "1", "2", "3", "4" второго "мешка".

7) `urn:oasis:names:tc:xacml:1.0:function:map`

В языке Haskell, эта функция определена следующим образом:

```
map:: (a -> b)  -> [a] -> [b]
map f []        = []
map f (x:xs)    = (f x) : (map f  xs)
```

В вышеуказанной системе обозначений, "f" является функцией, которую нужно применить, а "(x:xs)" представляет первый элемент списка, как "x", а оставшийся список, как "xs".

Например, следующее выражение,

```
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:map">
      <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-
normalize-to-lower-case">
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Hello</AttributeValue>
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">World!</AttributeValue>
      </Apply>
</Apply>
```

оценивается как "мешок", в котором содержатся "hello" и "world!".

БИБЛИОГРАФИЯ

[Haskell] THOMPSON (S.): Haskell: The Craft of Functional Programming (2nd Edition), *Addison Wesley*, ISBN 0-201-34275-8, 1996.

[IEEE 754] IEEE 754-1985, *Binary Floating-Point Arithmetic*, ISBN 1-5593-7653-8, IEEE Product No. SH10116-TBR.

[RBAC] ANSI INCITS 359-2004, *Information technology – Role Based Access Control,* http://csrc.nist.gov/rbac/.

СЕРИИ РЕКОМЕНДАЦИЙ МСЭ-Т

Серия A	Организация работы МСЭ-Т
Серия D	Общие принципы тарификации
Серия E	Общая эксплуатация сети, телефонная служба, функционирование служб и человеческие факторы
Серия F	Нетелефонные службы электросвязи
Серия G	Системы и среда передачи, цифровые системы и сети
Серия H	Аудиовизуальные и мультимедийные системы
Серия I	Цифровая сеть с интеграцией служб
Серия J	Кабельные сети и передача сигналов телевизионных и звуковых программ и других мультимедийных сигналов
Серия K	Защита от помех
Серия L	Конструкция, прокладка и защита кабелей и других элементов линейно-кабельных сооружений
Серия M	Управление электросвязью, включая СУЭ и техническое обслуживание сетей
Серия N	Техническое обслуживание: международные каналы передачи звуковых и телевизионных программ
Серия O	Требования к измерительной аппаратуре
Серия P	Качество телефонной передачи, телефонные установки, сети местных линий
Серия Q	Коммутация и сигнализация
Серия R	Телеграфная передача
Серия S	Оконечное оборудование для телеграфных служб
Серия T	Оконечное оборудование для телематических служб
Серия U	Телеграфная коммутация
Серия V	Передача данных по телефонной сети
Серия X	**Сети передачи данных, взаимосвязь открытых систем и безопасность**
Серия Y	Глобальная информационная инфраструктура, аспекты протокола Интернет и сети последующих поколений
Серия Z	Языки и общие аспекты программного обеспечения для систем электросвязи

Отпечатано в Швейцарии
Женева, 2009 г.