ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 55022— 2012

Информационная технология

СПЕЦИФИКАЦИЯ ЯЗЫКА ОПИСАНИЯ ПРЕДСТАВЛЕНИЯ ЗАДАЧ (JSDL). ВЕРСИЯ 1.0

Издание официальное

Москва
Стандартинформ
2019

gost.e-ecolog.ru

Предисловие

1 РАЗРАБОТАН Учреждением Российской академии наук Институтом радиотехники и электроники им. В. А. Котельникова РАН и Обществом с ограниченной ответственностью «Информационно-аналитический вычислительный центр» (ООО «ИАВЦ»)

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 22 «Информационные технологии»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 24 сентября 2012 г. № 425-ст

4 ВВЕДЕН ВПЕРВЫЕ

5 ПЕРЕИЗДАНИЕ. Январь 2019 г.

Правила применения настоящего стандарта установлены в статье 26 Федерального закона от 29 июня 2015 г. № 162-ФЗ «О стандартизации в Российской Федерации». Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (www.gost.ru)

© Стандартинформ, оформление, 2016, 2019

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

Содержание

1 Область применения . . . 1
2 Нормативные ссылки . . . 1
3 Термины, определения и сокращения . . . 1
4 Основные положения . . . 2
4.1 Спецификация JSDL . . . 2
4.2 Соглашения, принятые в настоящем стандарте . . . 3
5 Область действия JSDL . . . 4
5.1 О языке описания потребностей в ресурсах (RRL) . . . 5
5.2 О языке описания планирования (SDL) . . . 5
5.3 О соглашении о web-сервисах (WS-AG) . . . 5
5.4 О языке стратегии задач (JPL) . . . 5
5.5 О языке управления жизненным циклом задач (JLML) . . . 5
5.6 О потоке задач . . . 5
6 Структура документа JSDL . . . 6
7 Типы элементов JSDL . . . 6
7.1 Типы нормативной XML-схемы . . . 6
7.2 Типы языка JSDL . . . 6
7.2.1 Тип ProcessorArchitectureEnumeration . . . 7
7.2.2 Тип FileSystemTypeEnumeration . . . 7
7.2.3 Тип OperatingSystemTypeEnumeration . . . 7
7.2.4 Тип CreationFlagEnumeration . . . 8
7.2.5 Тип RangeValue_Type . . . 8
8 Базовый набор элементов JSDL . . . 9
8.1 Элементы структуры задачи . . . 10
8.1.1 Элемент JobDefinition . . . 10
8.1.2 Элемент JobDescription . . . 10
8.1.3 Элемент Description . . . 10
8.2 Элементы идентификации задачи . . . 11
8.2.1 Элемент JobIdentification . . . 11
8.2.2 Элемент JobName . . . 11
8.2.3 Элемент JobAnnotation . . . 11
8.2.4 Элемент JobProject . . . 12
8.3 Элементы приложения . . . 12
8.3.1 Элемент Application . . . 12
8.3.2 Элемент ApplicationName . . . 13
8.3.3 Элемент ApplicationVersion . . . 13
8.4 Элементы ресурсов . . . 13
8.4.1 Элемент Resources . . . 13
8.4.2 Элемент CandidateHosts . . . 14
8.4.3 Элемент HostName . . . 15
8.4.4 Элемент FileSystem . . . 15
8.4.5 Элемент MountPoint . . . 17
8.4.6 Элемент MountSource . . . 18
8.4.7 Элемент DiskSpace . . . 19

8.4.8 Элемент FileSystemType . . . 19
8.4.9 Элемент ExclusiveExecution . . . 20
8.4.10 Элемент OperatingSystem . . . 20
8.4.11 Элемент OperatingSystemType . . . 20
8.4.12 Элемент OperatingSystemName . . . 21
8.4.13 Элемент OperatingSystemVersion . . . 21
8.4.14 Элемент CPUArchitecture . . . 21
8.4.15 Элемент CPUArchitectureName . . . 22
8.4.16 Элемент IndividualCPUSpeed . . . 22
8.4.17 Элемент IndividualCPUTime . . . 23
8.4.18 Элемент IndividualCPUCount . . . 23
8.4.19 Элемент IndividualNetworkBandwidth . . . 24
8.4.20 Элемент IndividualPhysicalMemory . . . 24
8.4.21 Элемент IndividualVirtualMemory . . . 25
8.4.22 Элемент IndividualDiskSpace . . . 25
8.4.23 Элемент TotalCPUTime . . . 26
8.4.24 Элемент TotalCPUCount . . . 26
8.4.25 Элемент TotalPhysicalMemory . . . 27
8.4.26 Элемент TotalVirtualMemory . . . 27
8.4.27 Элемент TotalDiskSpace . . . 28
8.4.28 Элемент TotalResourceCount . . . 28
8.4.29 Дополнительные ресурсы . . . 29
8.5 Элементы потоков данных . . . 29
8.5.1 Элемент DataStaging . . . 29
8.5.2 Элемент FileName . . . 31
8.5.3 Элемент FilesystemName . . . 31
8.5.4 Элемент CreationFlag . . . 32
8.5.5 Элемент DeleteOnTermination . . . 32
8.5.6 Элемент Source . . . 33
8.5.7 Элемент URI . . . 34
8.5.8 Элемент Target . . . 34
9 Расширения JSDL . . . 35
9.1 Расширения атрибутов . . . 35
9.2 Расширения элементов . . . 35
9.3 Семантика значения «other» языка JSDL . . . 36
10 Нормативные расширения JSDL . . . 36
10.1 Выполнение задач на POSIX-совместимых узлах . . . 36
10.1.1 Элемент POSIXApplication . . . 36
10.1.2 Элемент Executable . . . 37
10.1.3 Элемент Argument . . . 38
10.1.4 Элемент Input . . . 39
10.1.5 Элемент Output . . . 40
10.1.6 Элемент Error . . . 40
10.1.7 Элемент WorkingDirectory . . . 41
10.1.8 Элемент Environment . . . 41
10.1.9 Элемент WallTimeLimit . . . 42

10.1.10 Элемент FileSizeLimit . . . 42
10.1.11 Элемент CoreDumpLimit . . . 42
10.1.12 Элемент DataSegmentLimit . . . 43
10.1.13 Элемент LockedMemoryLimit . . . 43
10.1.14 Элемент MemoryLimit . . . 43
10.1.15 Элемент OpenDescriptorsLimit . . . 44
10.1.16 Элемент PipeSizeLimit . . . 44
10.1.17 Элемент StackSizeLimit . . . 44
10.1.18 Элемент CPUTimeLimit . . . 44
10.1.19 Элемент ProcessCountLimit . . . 45
10.1.20 Элемент VirtualMemoryLimit . . . 45
10.1.21 Элемент ThreadCountLimit . . . 45
10.1.22 Элемент UserName . . . 46
10.1.23 Элемент GroupName . . . 46
11 Вопросы безопасности . . . 46
Приложение А (обязательное) Нормативная схема JSDL . . . 47
Приложение Б (обязательное) Нормативная схема JSDL для приложений POSIX . . . 54
Приложение В (обязательное) Нормативная схема JSDL, содержащая сводку исправлений . . . 57
Приложение Г (справочное) Расширенные информационные примеры JSDL . . . 58

Введение

Язык описания представления задач (JSDL) представляет собой язык описания требований вычислительных задач к ресурсам, особенно в Грид-среде, хотя и не ограничивается ими. Язык JSDL содержит словарь и нормативную XML-схему, которые упрощают описание этих требований, используя набор элементов XML.

Версия JSDL 1.0 дает базовый словарь описания задачи для ее представления в Грид-среде. Этот базовый словарь включает в себя понятия, используемые в ряде существующих систем: Condor, Globus Toolkit, Load Sharing Facility (LSF), Portable Batch System (PBS), (Sun) GridEngine (SGE), Uniform Interface to Computing Resources (Unicore).

Предполагается, что реализация более сложных функций будет обеспечена за счет нестандартизованного расширения версии JSDL 1.0. Часть таких функций может стать нормативными стандартными расширениями языка в установленном порядке.

Элементы JSDL подразделяются на три следующие общие категории:

- требования к идентификации задачи;
- требования к ресурсам;
- требования к данным.

Элементы языка JSDL версии 1.0 ограничены описанием требований задач на момент представления. Не существует элементов, определенных в спецификации JSDL версии 1.0 таким образом, чтобы они содержали информацию о задаче, после того как она была представлена. Такая информация, как уникальные идентификаторы задач или информация о статусе задачи, как правило, поддерживается базовыми системами управления задачами. Метод доступа к такого сорта информации может быть описан в отдельных документах либо добавлен в спецификацию JSDL версии 1.0 в качестве расширений.

Раздел 5 дает обоснование области применимости JSDL 1.0. Раздел 6 содержит описание структуры документа, написанного на языке JSDL. Типы элементов языка и их базовый набор приведены в разделах 7 и 8 соответственно. JSDL имеет расширяемую спецификацию. Правила создания расширений языка JSDL и описание нормативных расширений языка содержатся в разделах 9 и 10. Вопросы безопасности обсуждаются в разделе 11. Нормативная XML-схема приведена в приложении А.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационная технология

СПЕЦИФИКАЦИЯ ЯЗЫКА ОПИСАНИЯ ПРЕДСТАВЛЕНИЯ ЗАДАЧ (JSDL). ВЕРСИЯ 1.0

Information technology.
Job submission description language (JSDL) specification.
Version 1.0

Дата введения — 2014—01—01

1 Область применения

В настоящем стандарте приведены семантика и структура языка описания представления задач (JSDL), который предназначен для описания требований, предъявляемых вычислительными задачами к ресурсам, в большей степени к Грид-системам, хотя и не ограничивается только ими. Использование языка JSDL позволяет обеспечить интероперабельность как на уровне взаимодействия пользователей с различными вычислительными системами, так и на уровне взаимодействия вычислительных систем между собой.

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие стандарты:

ГОСТ Р 1.1 Стандартизация в Российской Федерации. Технические комитеты. Правила создания и деятельности

ГОСТ Р 1.12 Стандартизация в Российской Федерации. Термины и определения

Примечание — При пользовании настоящим стандартом целесообразно проверить действие ссылочных стандартов в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет или по ежегодному информационному указателю «Национальные стандарты», который опубликован по состоянию на 1 января текущего года, и по выпускам ежемесячного информационного указателя «Национальные стандарты» за текущий год. Если заменен ссылочный стандарт, на который дана недатированная ссылка, то рекомендуется использовать действующую версию этого стандарта с учетом всех внесенных в данную версию изменений. Если заменен ссылочный стандарт, на который дана датированная ссылка, то рекомендуется использовать версию этого стандарта с указанным выше годом утверждения (принятия). Если после утверждения настоящего стандарта в ссылочный стандарт, на который дана датированная ссылка, внесено изменение, затрагивающее положение, на которое дана ссылка, то это положение рекомендуется применять без учета данного изменения. Если ссылочный стандарт отменен без замены, то положение, в котором дана ссылка на него, рекомендуется применять в части, не затрагивающей эту ссылку. Сведения о действии сводов правил целесообразно проверить в Федеральном информационном фонде технических регламентов и стандартов.

3 Термины, определения и сокращения

3.1 В настоящем стандарте применены термины по ГОСТ Р 1.1, ГОСТ Р 1.12, а также следующие термины с соответствующими определениями:

3.1.1 **производящая система:** Система, генерирующая документ, написанный на языке JSDL.

3.1.2 **потребляющая система:** Система, обрабатывающая документ, написанный на языке JSDL.

3.1.3 **документ JSDL:** Документ, написанный на языке JSDL.

3.1.4 **множественность элемента:** Допустимое число экземпляров элемента в одном документе, написанном на языке JSDL.

Издание официальное

3.2 В настоящем стандарте также применены следующие сокращения:

JSDL (Job Submission Description Language) — язык описания представления задач;
XML (Extensible Markup Language) — расширяемый язык разметки;
RRL (Resource Requirements Language) — язык описания потребностей в ресурсах;
SDL (Scheduling Description Language) — язык описания планирования;
JPL (Job Policy Language) — язык стратегии задач;
JLML (Job Lifetime Management Language) — язык управления жизненным циклом задач;
WS-AG (Web Service Agreement) — соглашение о web-сервисах;
ISA (Instruction Set Architecture) — архитектура набора инструкций;
CIM (Common Information Model) — общая информационная модель;
NFS (Network File System) — сетевая файловая система;
POSIX (Portable Operating System Interface) — переносимый интерфейс операционных систем.

4 Основные положения

4.1 Спецификация JSDL

Спецификация языка мотивирована двумя основными сценариями.

Во-первых, многие организации, разрабатывающие Грид-системы, устанавливают различные системы управления задачами. При этом каждая система имеет свой собственный язык для описания требований представления задач. Это делает взаимодействие между этими системами управления задачами сложным. Для того чтобы использовать такие различные системы управления задачами, организации вынуждены подготовить и поддерживать целый ряд различных документов, касающихся представления задач, — по одному для каждой системы, каждый из которых описывает одно и то же представление. Стандартизованный язык, такой, например, как JSDL, может быть легко перенесен на различные системы и тем самым может полностью решить эту проблему.

Во-вторых, Грид-среда, как заведомо гетерогенная, вовлекает во взаимодействие целый ряд систем управления заданиями различных типов. На рисунке 1 проиллюстрирован один из возможных видов такой Грид-среды. Описание представления задачи (показано в виде документа JSDL на рисунке 1) в Грид-среде может быть преобразовано посредниками или дополнительно уточнено при помощи информации, первоначально отсутствовавшей у первоначального заявителя задачи.

Это описание может быть передано между системами или реализовано на одном из ресурсов, удовлетворяющем требованиям данной задачи. Все эти взаимодействия могут осуществляться автоматически при помощи стандартного языка, который может быть легко транслирован на собственный язык каждой из систем.

На рисунке 1 в качестве примера показана часть систем, в которых язык JSDL может быть использован в Грид-среде. Использование JSDL не ограничивается только этими системами. Может существовать множество других потребителей, к которым могут относиться: системы учета, системы защиты, системы архивации, системы аудита и пр.

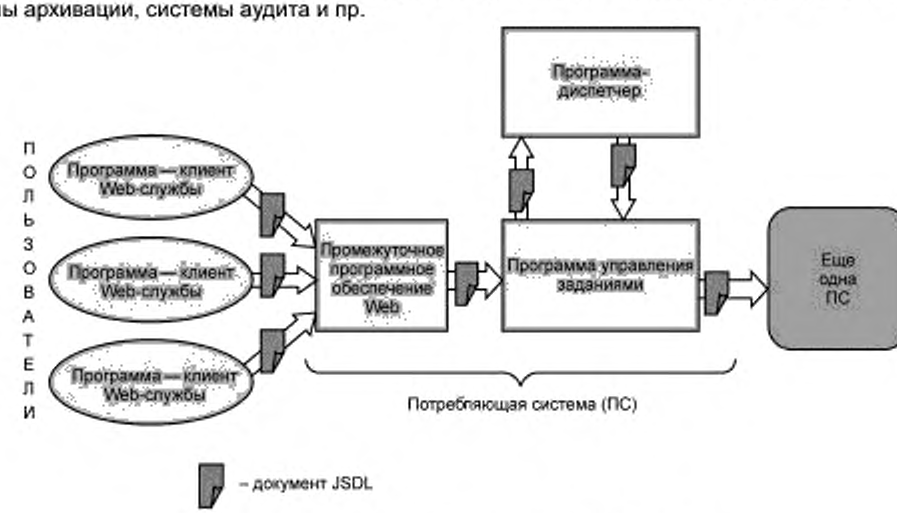


Рисунок 1 — Пользователи JSDL в Грид-среде

4.2 Соглашения, принятые в настоящем стандарте

Ключевые слова «обязан», «недопустимо», «требуемый», «должен», «не позволяется», «рекомендуется», «может» и «опциональный» интерпретируются следующим образом:

- «обязан» — это слово, а также термины «требуется» и «нужно» используются для требований, которые являются абсолютно необходимыми в данной спецификации;
- «недопустимо» — это слово и фраза «не позволяется» означают абсолютный запрет в рамках спецификации;
- «требуемый» — это слово используется для требований, которые являются абсолютно необходимыми в данной спецификации;
- «должен» — это слово используется для требований, которые являются абсолютно необходимыми в данной спецификации;
- «не позволяется» — эта фраза означает абсолютный запрет в рамках спецификации;
- «рекомендуется» — это слово используется для обозначения требований, от выполнения которых можно отказаться при наличии разумных причин. Однако при таком отказе следует помнить о возможных проблемах в результате отказа и принимать взвешенное решение;
- «может» — это слово используется для обозначения требований, от выполнения которых можно отказаться при наличии разумных причин. Однако при таком отказе следует помнить о возможных проблемах в результате отказа и принимать взвешенное решение;
- «опциональный» — это слово обозначает элементы, реализация которых является необязательной. Одни разработчики могут включать такие опции в свою продукцию для расширения возможностей, а другие — опускать в целях упрощения. Реализация, не включающая ту или иную опцию, должна быть готова к работе с реализациями, которые используют эту опцию (возможно, совместная работа будет обеспечиваться за счет некоторого ущерба функциональности). Включающие опцию реализации должны быть готовы (естественно, без использования такой опции) к взаимодействию с реализациями, которые такую опцию не поддерживают.

Для каждого компонента перед его описанием приводятся псевдосхемы. Псевдосхемы используют следующие соглашения для описания атрибутов и элементов: '?' означает наличие одного экземпляра объекта либо его отсутствие; '*' означает либо отсутствие экземпляра, либо их произвольное количество; '+' означает наличие одного или большего числа экземпляров. Обычно атрибутам присваивается значение, соответствующее их типу, как это определено в нормативной схеме:

```
<!--пример псевдосхемы -->
<defined_element
required_attribute_of_type_string="xsd:string"
optional_attribute_of_type_int="xsd:int"? >
<required_element />
<optional_element />?
<one_or_more_of_this_element />+
</defined_element>
```

Спецификация на всем протяжении использует префиксы пространства имен, которые перечислены в таблице 4.1. Следует заметить, что выбор префикса пространства имен произволен и семантически не существенен.

Таблица 4.1 — Префиксы и пространства имен, используемые в этой спецификации

Префикс	Пространство имен
xsd	gost.e-ecolog.ru
jsdl	gost.e-ecolog.ru /jsdl
jsdl-posix	gost.e-ecolog.ru
jsdl-errata	gost.e-ecolog.ru

Термины «элемент JSDL» и «атрибут JSDL» указывают на то, что соответствующая языковая конструкция представлена в нормативной схеме JSDL либо как элемент языка XML, либо как атрибут языка XML в нормативной схеме JSDL.

Термин «документ JSDL» означает правильно отформатированный документ XML, который может быть проверен на соответствие нормативному определению схемы JSDL, которое содержится в приложении А.

Ключевое слово «present», использованное для ссылки на элемент JSDL, подразумевает, что экземпляр элемента содержится в документе JSDL. Ключевое слово «support» по отношению к потребительской системе, использующей JSDL и поддерживающей спецификацию JSDL и языковые конструкции, относится к возможности этой системы проводить синтаксический анализ документа JSDL, т. е. потребительская система должна иметь возможность интерпретировать языковые конструкции и присваивать им семантические конструкции, описанные в этой спецификации, и присваивать им значения, содержащиеся в документе JSDL. Таким образом, все потребляющие системы, совместимые с JSDL, должны поддерживать все языковые конструкции JSDL. Ключевое слово «satisfy» по отношению к потребительской системе, удовлетворяющей документу JSDL, означает, что система может как поддерживать, так и удовлетворять все элементы, представленные в документе JSDL. Элемент считается удовлетворенным, если потребительская система может обеспечить надлежащее выполнение значения, присвоенного элементу в документе JSDL.

5 Область действия JSDL

JSDL — язык, используемый для описания требований представления индивидуальных задач. Язык не пытается описывать полный жизненный цикл задачи или взаимодействие между отдельными задачами. Таким образом, JSDL допускает, что процессы управления и исполнения задач, описанных и представленных с использованием JSDL, в частности в Грид-среде, могут требовать ряд других языков и протоколов. Например, как для описания взаимодействия отдельных задач, описанных при помощи JSDL, так и для описания взаимодействия этих задач с потребляемыми или производимыми ими данными следует использовать язык рабочего потока.

В качестве другого примера рассмотрим процесс переговоров, который необходим при размещении задачи в оптимальной среде ресурсов в соответствии с ее требованиями. В этом случае требуется использование удовлетворительного процесса переговоров и согласованных протоколов. Примером такого протокола является протокол, который определен в спецификации GGF WS-соглашения (WS-AG). На рисунке 2 приведены некоторые другие элементы, необходимые для представления, управления и выполнения задач, особенно для Грид-систем. Многие из этих элементов являются языками и протоколами, которые еще отсутствуют, либо только разрабатываются в настоящее время, либо требуют создания спецификаций и проведения стандартизации.

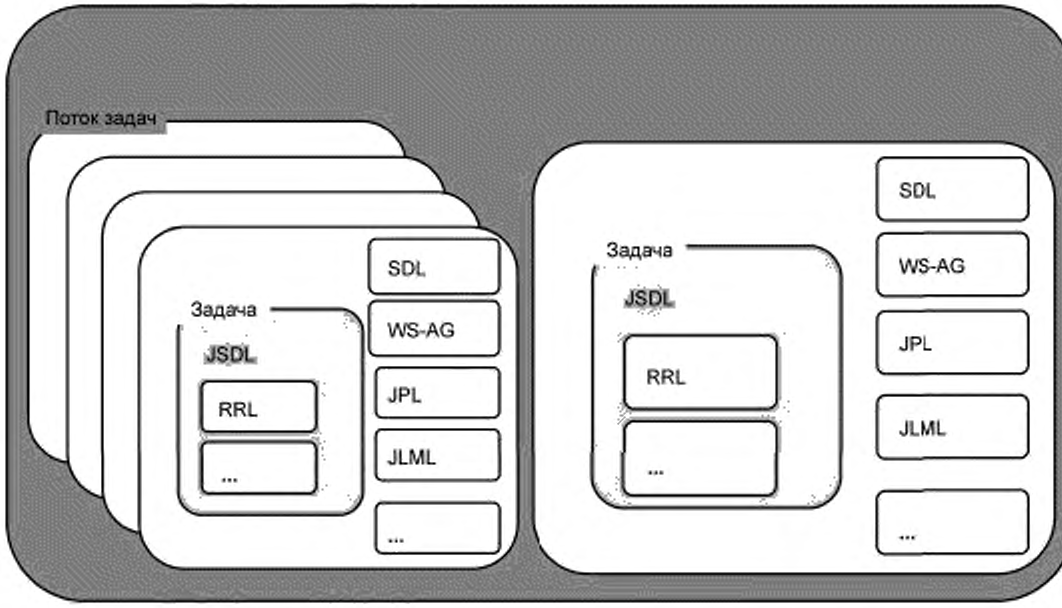


Рисунок 2 — Взаимодействие между JSDL и другими спецификациями

5.1 О языке описания потребностей в ресурсах (RRL)

Спецификация JSDL могла бы использовать язык для описания потребностей задачи в ресурсах (RRL). Определение RRL должно быть дано в своей спецификации и поэтому выходит за рамки спецификации JSDL. Однако в отсутствие подходящего языка RRL здесь задается основной набор элементов, которые должны содержаться в RRL. Вполне вероятно, что будущие версии спецификации JSDL будут использовать стандартные спецификации RRL (если будут доступны).

5.2 О языке описания планирования (SDL)

Планирование задач и описание требований планирования являются комплексными проблемами. Требования планирования распределяются по многим категориям. Требования включают в себя:

- планирование времени. Время начала и окончания задачи, срок действия окон для исполнения задачи и т. д.;
- расписание, зависящее от данных. Планирование задач в зависимости от ожидаемого наличия входных данных и т. д.;
- планирование, зависящее от рабочего потока. Планирование задач в зависимости от порядка задач в рабочем потоке.

Кроме того, могут существовать зависимости между этими категориями в результате планирования требуемых ресурсов задач. Например, может существовать зависимость между заданным порядком задач в рабочем потоке и входными данными, необходимыми для выполнения запланированной задачи. Из-за сложности описания требований расписания данная проблема вынесена за рамки языка JSDL.

5.3 О соглашении о web-сервисах (WS-AG)

Разработка соглашения между ресурсными web-сервисами переднего плана и web-сервисами представления задач в настоящее время лежит в рамках спецификации GGF WS-соглашения (WS-AG).

WS-соглашения рассматривают как важный новый стандарт представления ресурсов задачи. Однако многие используемые в настоящий момент системы не имеют web-сервисов переднего плана и могут не потребовать такого рода функциональности. Таким образом, важно, чтобы JSDL мог быть использован как совместно с другими стандартами, так и самостоятельно. Другими словами, JSDL может быть использован как с WS-AG, так и без него.

5.4 О языке стратегии задач (JPL)

Стратегия задач является очень широкой темой. Стратегии могут применяться во многих аспектах выполнения задачи. Пользователи могут хотеть применять стратегии для задач, собственники ресурсов также могут хотеть использовать стратегии для использования задачами своих ресурсов. Описание таких ограничений в стратегиях требует своего собственного языка описания и лежит вне рамок JSDL. Язык описания стратегий может рассматриваться как родственный JSDL.

5.5 О языке управления жизненным циклом задач (JLML)

Управление жизненным циклом задач является важным аспектом общего управления задачами. Это управление позволяет описывать состояния задач и директивы управления задачами. Очевидна необходимость разработки языка работы управления жизненным циклом задач, который мог бы описывать этот аспект управления задачами. Так как управление жизненным циклом задач связано с работой после представления задачи, язык управления жизненным циклом задач выходит за рамки JSDL.

5.6 О потоке задач

Язык JSDL предназначен для описания шаблонов задач при представлении отдельных задач на соответствующих вычислительных ресурсах. Следует учитывать существование большой заинтересованности в развитии языков для описания параметрических задач, совместного распределения на вычислительных ресурсах, потоков задач и т. д. Такие языки могут «знать о существовании JSDL», другими словами, они могут ссылаться на описания отдельных задач, выполненных на языке JSDL, для каждого рабочего узла. Очевидно, однако, что определение такого языка представляет собой отдельное направление деятельности, и может быть создана надстройка к стандарту JSDL, что, следовательно, выходит за рамки JSDL.

6 Структура документа JSDL

Документ JSDL описывается с использованием языка XML и связан с нормативной схемой XML, содержащейся в приложении А.

Документ JSDL организован следующим образом: корневой элемент JobDefinition содержит единственный обязательный дочерний элемент JobDescription. Элемент JobDescription содержит элементы, описывающие задачу: JobIdentification, Application, Resources, DataStaging. Определение псевдосхемы следующее:

```
<JobDefinition>
    <JobDescription>
        <JobIdentification ... />?
        <Application ... />?
        <Resources ... />?
        <DataStaging ... />*
        </JobDescription>
    <xsd:any##other/>*
    </JobDefinition>
```

Полные примеры документов JSDL приведены в приложении В.

Спецификация JSDL не определяет значения по умолчанию для элементов, которые отсутствуют в документе JSDL. [Типичные значения для таких элементов присваиваются системой потребления задач (потребляющей системой). Для получения более подробной информации следует обратиться к спецификации каждого элемента.] Предполагается, что элементы JSDL, требующие описания для конкретного представления задачи, будут представлены совместно с соответствующими значениями в документе JSDL.

Все элементы, присутствующие в документе JSDL, должны быть выполнены для всего документа, который будет удовлетворен системой потребления. Конечно, существует возможность создания документов JSDL, которые не могут быть удовлетворены. Например, документ JSDL, который содержит противоречивые потребности в ресурсах, не может быть удовлетворен. Однако следует отметить, что результат представления документа JSDL на потребляющую систему спецификацией JSDL не рассматривается.

7 Типы элементов JSDL

Спецификация JSDL использует некоторое число типов нормативных XML-схем. Она также использует ряд типов, специфичных для описания потребностей задач. Группа JSDL-WG основывается на определении ряда типов, приведенных в уже существующих стандартах, в частности на стандарте Общей информационной модели (CIM) и Интерфейсе переносимых операционных систем (POSIX).

Поскольку не существует определения нормативных XML-схем для этих типов, они представлены в этом документе. Типы операционных систем (см. 7.2.3) определены, основываясь на CIM.

Некоторое число типов определяют специальное значение «other». Это значение может быть использовано для введения элементов из других спецификаций (детально см. в 9.3).

7.1 Типы нормативной XML-схемы

Спецификация JSDL принимает типы нормативных XML-схем (xsd), приведенные в таблице 7.1.

Таблица 7.1 — Типы нормативных XML-схем

xsd:string	xsd:normalizedString	xsd:nonNegativeInteger	xsd:boolean
xsd:NCName	xsd:anyType	xsd:token	xsd:any##other
Complex			

7.2 Типы языка JSDL

В спецификации JSDL версии 1.0 определены следующие типы:

- ProcessorArchitectureEnumeration;
- FileSystemTypeEnumeration;

- OperatingSystemTypeEnumeration;
- CreationFlagEnumeration;
- RangeValue_Type.

7.2.1 Тип ProcessorArchitectureEnumeration

Перечисляемый тип архитектур процессоров основан на именах малого числа общих архитектур процессоров, определенных в ISA. Значения типа, приведенные в таблице 7.2, должны поддерживаться.

Таблица 7.2 — Архитектуры процессоров (jsdl:ProcessorArchitectureEnumeration)

Нормативное имя JSDL	Определение
spark	Процессор архитектуры SPARC
powerpc	Процессор архитектуры PowerPC
x86	Процессор архитектуры Intel, основанный на наборе микросхем 8086
x86_32	Процессор x86, работающий в 32-битовом режиме
x86_64	Процессор x86, работающий в 64-битовом режиме
parisc	Процессор архитектуры PARISC
mips	Процессор архитектуры MIPS
ia64	Процессор архитектуры Intel 64 бит
arm	Процессор ARM
other	Значение, не определенное в этом списке

Данный перечень типов является расширяемым. Расширения этого списка не должны содержаться в пространстве имен JSDL.

7.2.2 Тип FileSystemTypeEnumeration

Типы файловых систем, перечисленные в таблице 7.3, должны поддерживаться.

Таблица 7.3 — Типы файловых систем (jsdl:FileSystemTypeEnumeration)

Нормативное имя JSDL	Определение
swap	Стандартное пространство подкачки для подкачки памяти
temporary	Стандартное временное пространство для файлов, которые периодически удаляются. Это пространство оказывается недоступным после окончания задачи
spool	Временное пространство, которое может сохраняться после окончания задачи
normal	Нормальное пространство для файлов, которые можно читать и/или в которые можно писать. Файлы не удаляются после окончания задачи до тех пор, пока сам пользователь не запросит их удаления

Этот список типов является расширяемым. Расширение списка не должно быть в пространстве имен JSDL.

7.2.3 Тип OperatingSystemTypeEnumeration

Типы, перечисленные в таблице 7.4, должны поддерживаться потребляющей системой. Значения базируются на значениях поля OSType записи CIM_OperatingSystem модели CIM.

Таблица 7.4 — Типы операционных систем (jsdl:OperatingSystemTypeEnumeration)

Нормативные имена JSDL			
Unknown	WINNT	LINUX	HP_MPE
Other	WINCE	Lynx	NextStep
MACOS	NCR3000	XENIX	PalmPilot
ATTUNIX	NetWare	VM	Rhapsody
DGUX	OSF	Interactive_UNIX	Windows_2000
DECNT	DC_OS	BSDUNIX	Dedicated
Tru64_UNIX	Reliant_UNIX	FreeBSD	OS_390
OpenVMS	SCO_UnixWare	NetBSD	VSE
HPUX	SCO_OpenServer	GNU_Hurd	TPF
AIX	Sequent	OS9	Windows_R_Me
MVS	IRIX	MACH_Kernel	Caldera_Open_UNIX
OS400	Solaris	Inferno	OpenBSD
OS_2	SunOS	QNX	Not_Applicable
JavaVM	U6000	EPOC	Windows_XP
MSDOS	ASERIES	IxWorks	z_OS
WIN3x	TandemNSK	VxWorks	
WIN95	TandemNT	MiNT	
WIN98	BS2000	BeOS	

Данный список типов является расширяемым. Расширение списка не должно быть в пространстве имен JSDL.

7.2.4 Тип CreationFlagEnumeration

Должны поддерживаться флаги режима создания файлов, приведенные в таблице 7.5.

Таблица 7.5 — Флаги режима создания файлов (jsdl:CreationFlagEnumeration)

Нормативное имя JSDL	Определение
overwrite	Переписать существующий файл с тем же именем или создать новый
dontOverwrite	Создать файл, если он не существует; в противном случае использовать существующий, не переписывая его
append	Добавить к существующему файлу с тем же именем; в противном случае создать новый

7.2.5 Тип RangeValue_Type

Значение диапазона является комплексным типом, который позволяет определять точные величины (с опциональным аргументом «эпсилон»), интервалы, открытые слева или справа, и диапазоны. Все данные числа имеют тип xsd:double. UpperBoundedRanges and LowerBoundedRanges задают верхнюю и нижнюю границы соответственно. Диапазоны могут быть «неограниченными» в смысле либо отрицательной, либо положительной бесконечности, в зависимости от возможностей потребляющей системы. Например, применительно к Java, «бесконечность» может выражаться как java.lang.Double. NEGATIVE_INFINITY и java.lang.Double.POSITIVE_INFINITY соответственно.

Если опциональный атрибут exclusiveBound не определен, то он имеет значение «false».

Если опциональный атрибут «epsilon» не определен, то он имеет значение 0, и потребляющая система должна использовать точное совпадение, отвечающее максимально возможной точности представления числа на данной потребляющей системе. Если атрибут «epsilon» определен, то его величина определяет допустимую точность сравнения данного значения в отличие от элемента «Exact». Таким образом, элемент «Exact» при заданном значении атрибута «epsilon» определяет диапазон «[значение-epsilon, значение+epsilon]». Значение «epsilon» не должно быть отрицательным.

Значения диапазона величин, определяющие пересекающиеся диапазоны, могут быть стянуты вместе потребляющей системой, для того чтобы соответствовать описанию задачи, однако при этом документ JSDL не должен быть изменен.

Этот тип обязан поддерживаться потребляющей системой. Формальная семантика совпадений определена в следующем разделе (7.2.5.1).

7.2.5.1 Семантика совпадений

Должна использоваться следующая семантика совпадений:

- численное значение N попадает в диапазон R тогда и только тогда, если выполнено хотя бы одно из следующих условий (имеет значение «true»):

- R содержит UpperBoundedRange U, атрибут exclusiveBound имеет значение «false» и $N \le U$;

- R содержит UpperBoundedRange U, атрибут exclusiveBound имеет значение «true» и $N < U$;

- R содержит LowerBoundedRange L, атрибут exclusiveBound имеет значение «false» и $N \ge L$;

- R содержит LowerBoundedRange L, атрибут exclusiveBound имеет значение «true» и $N > L$;

- R содержит «Exact» E с атрибутом «epsilon», равным ε, и $E - \varepsilon \le N \le E + \varepsilon$;

- R содержит диапазон Range с LowerBound L и UpperBound U, и следующие два выражения истинны:

- L имеет атрибут exclusiveBound, равный «false», и $N \ge L$ либо L имеет атрибут exclusiveBound, равный «true», и $N > L$,

- U имеет атрибут exclusiveBound, равный «false», и $N \le L$ либо U имеет атрибут exclusiveBound, равный «true», и $N < L$.

7.2.5.2 Псевдосхема

```
<UpperBoundedRange exclusiveBound="xsd:boolean"?>
        xsd:double
</UpperBoundedRange>?
<LowerBoundedRange exclusiveBound="xsd:boolean"?>
        xsd:double
</LowerBoundedRange>?
<Exact epsilon="xsd:double"?>
        xsd:double
</Exact>*
<Range>
        <LowerBound exclusiveBound="xsd:boolean"?>
        xsd:double
</LowerBound>
<UpperBound exclusiveBound="xsd:boolean"?>
        xsd:double
</UpperBound>
</Range>*
```

7.2.5.3 Примеры

Псевдовыражение «5, 6.7777, 7.0, [50.3,99.5), [100-», которое определяет неперекрывающийся диапазон, содержащий следующие значения: 5, 6.7777, 7.0, все величины от 50.3 включительно до 99.5, не включая последнее, и все величины от 100 и более, может быть описано при помощи RangeValue следующим образом:

```
<jsdl:LowerBoundedRange> 100.0 </jsdl:LowerBoundedRange>
<jsdl:Exact> 5.0 </jsdl:Exact>
<jsdl:Exact epsilon="0.00001"> 6.7777 </jsdl:Exact>
<jsdl:Exact> 7.0 </jsdl:Exact>
<jsdl:Range>
<jsdl:LowerBound> 50.3 </jsdl:LowerBound>
<jsdl:UpperBound exclusiveBound="true"> 99.5 </jsdl:UpperBound>
</jsdl:Range>
```

8 Базовый набор элементов JSDL

Базовый набор элементов JSDL содержит семантику элементов, определенных в JSDL 1.0. Все элементы должны поддерживаться потребляющими системами, удовлетворяющими требованиям JSDL 1.0 (см. раздел 2).

8.1 Элементы структуры задачи

8.1.1 Элемент JobDefinition

8.1.1.1 Определение

Этот элемент описывает задачу и ее требования. Он содержит секцию JobDescription и является корневым элементом документа JSDL.

8.1.1.2 Множественность

Множественность этого элемента равна единице.

8.1.1.3 Тип

Это комплексный тип. Он должен поддерживать следующие элементы:

JobDescription.

8.1.1.4 Атрибуты

Определены следующие атрибуты:

id — идентификатор документа определения задачи. Он определен как "xsd:ID" и находится в стандартном пространстве имен документа. Атрибут может быть опущен.

8.1.1.5 Псевдосхема

```
<JobDefinition id="xsd:ID"?>
<JobDescription ... />
<xsd:any##other/>*
</JobDefinition>
```

8.1.1.6 Примеры

Документ с идентификатором "gnuplot":

```
<jsdl:JobDefinition id="gnuplot">
<jsdl:JobDescription> ... </jsdl:JobDescription>
</jsdl:JobDefinition>
```

Пример приведен в приложении Г.

8.1.2 Элемент JobDescription

8.1.2.1 Определение

Данный элемент описывает задачу и ее требования. Он содержит следующие элементы: JobIdentification, Application, Resources, DataStaging.

8.1.2.2 Множественность

Множественность этого элемента равна единице.

8.1.2.3 Тип

Это комплексный тип. Он должен поддерживать следующие элементы:

- JobIdentification;
- Application;
- Resources;
- DataStaging.

8.1.2.4 Атрибуты

Атрибуты не определены.

8.1.2.5 Псевдосхема

```
<JobDescription>
<JobIdentification ... />?
<Application ... />?
<Resources ... />?
<DataStaging ... />*
<xsd:any##other>*
</JobDescription>
```

8.1.3 Элемент Description

8.1.3.1 Определение

Данный элемент дает описательную, легко читаемую информацию о комплексном элементе, содержащимся в нем. Он может присутствовать как подэлемент целого ряда других элементов JSDL: JobIdentification, Application, FileSystem и др. Если этот элемент отсутствует как подэлемент, то описание не определено.

8.1.3.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.1.3.3 Тип

Тип этого элемента xsd:string.

8.1.3.4 Атрибуты

Атрибуты не определены.

8.1.3.5 Псевдосхема

<Description> xsd:string </Description>?

8.2 Элементы идентификации задачи

8.2.1 Элемент JobIdentification

8.2.1.1 Определение

Этот элемент содержит все элементы, идентифицирующие задачу: JobName, Description, JobAnnotation, JobProject. Если этот элемент не представлен, то его значение, включая значения всех его подэлементов, является неопределенным.

8.2.1.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.2.1.3 Тип

Это комплексный тип. Он должен поддерживать следующие элементы:

- JobName;
- Description;
- JobAnnotation;
- JobProject.

8.2.1.4 Атрибуты

Атрибуты не определены.

8.2.1.5 Псевдосхема

<JobIdentification>
<JobName ... />?
<Description ... />?
<JobAnnotation ... />*
<JobProject ... />*
<xsd:any##other>*
</JobIdentification>?

8.2.2 Элемент JobName

8.2.2.1 Определение

Этот элемент является строкой, которая может определяться пользователем для задания имени задачи, описываемой в документе JSDL. Имя может не быть уникальным для конкретного документа JSDL, т. е. пользователь может задавать одно и то же имя задачи во многих документах JSDL. Если этот элемент не существует, то он является неопределенным.

8.2.2.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.2.2.3 Тип

Тип элемента xsd:string.

8.2.2.4 Атрибуты

Атрибуты не определены.

8.2.2.5 Псевдосхема

<JobName> xsd:string </JobName>?

8.2.2.6 Примеры

Имя может быть использовано для поиска и сортировки задач, а также для управления задачами.
<jsdl:JobName> visualization </jsdl:JobName>

8.2.3 Элемент JobAnnotation

Данный элемент устарел и может быть удален в будущих версиях JSDL. Этот элемент не должен использоваться в разрабатываемых производящих системах.

8.2.3.1 Определение

Этот элемент является строкой, которая может определяться пользователем для описания задачи. Если этот элемент отсутствует, то его значение не определено. В отличие от элемента Description элемент JobAnnotation может содержать информацию, касающуюся потребляющей системы.

8.2.3.2 Множественность

Множественность этого элемента равна нулю и более.

8.2.3.3 Тип

Тип элемента xsd:string.

8.2.3.4 Атрибуты

Атрибуты не определены.

8.2.3.5 Псевдосхема

<JobAnnotation> xsd:string </JobAnnotation>*

8.2.3.6 Примеры

Аннотация может быть использована для задания уникального идентификатора, однозначно интерпретируемого потребляющей системой.

<jsdl:JobAnnotation> uuid:ashjasuwhqsoi </jsdl:JobAnnotation>

8.2.4 Элемент JobProject

8.2.4.1 Определение

Этот элемент является строкой, задающей проект, частью которого является задача. Проект может использоваться системами контроля доступа или системами учетных записей. Интерпретация элементов JobProject выполняется потребляющей системой. Если этот элемент отсутствует, то он является неопределенным.

8.2.4.2 Множественность

Множественность этого элемента равна нулю или более.

8.2.4.3 Тип

Тип элемента xsd:string.

8.2.4.4 Атрибуты

Атрибуты не определены.

8.2.4.5 Псевдосхема

<JobProject> xsd:string </JobProject>*

8.2.4.6 Примеры

Задача является частью проекта wcdma.

<jsdl:JobProject> wcdma </jsdl:JobProject>

8.3 Элементы приложения

Язык JSDL поддерживает только описание индивидуальных задач (см. раздел 7). Не более одного приложения может быть определено в документе JSDL.

8.3.1 Элемент Application

8.3.1.1 Определение

Этот элемент описывает приложение и его требования. Он содержит следующие элементы: ApplicationName, ApplicationVersion, Description. Он служит общим контейнером высокого уровня и предназначен для проведения более специфических определений приложения. Одним из таких определений является нормативное расширение, удовлетворяющее POSIX и данное в 10.1.1. Используемый без всяких расширений, элемент одинаковым образом описывает приложение при помощи имени и номера версии. Если элемент отсутствует, определение задачи не определено в исполняемом приложении. Документ JSDL может определить задачу с постановочными данными либо пустую задачу (см. 8.3.2 и 8.1.2).

8.3.1.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.3.1.3 Тип

Это комплексный тип. Он должен поддерживать следующие элементы:

- ApplicationName;
- ApplicationVersion;
- Description.

8.3.1.4 Атрибуты

Атрибуты не определены.

8.3.1.5 Псевдосхема

<Application>
<ApplicationName ... />?
<ApplicationVersion ... />?
<Description ... />?

<xsd:any##other/>*
</Application>?

8.3.2 Элемент ApplicationName

8.3.2.1 Определение

Этот элемент является строкой, которая задает имя приложения, используемое для идентификации приложения вне зависимости от его расположения на вычислительном узле или системе. Если элемент отсутствует, то он не определен и предполагается пустая (нулевая) задача, за исключением случая, когда присутствует элемент расширения приложения, который определяет исполняемое приложение (см. 8.3.1 и 8.1.2).

8.3.2.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.3.2.3 Тип

Тип элемента xsd:string.

8.3.2.4 Атрибуты

Атрибуты не определены.

8.3.2.5 Псевдосхема

<ApplicationName> xsd:string </ApplicationName>?

8.3.2.6 Примеры

Приложение BLAST:

<jsdl:ApplicationName> BLAST </jsdl:ApplicationName>

8.3.3 Элемент ApplicationVersion

8.3.3.1 Определение

Этот элемент является строкой, которая задает версию исполняемого приложения. Потребляющая система должна использовать точное текстовое совпадение для выбора версии приложения. Если этот элемент отсутствует, то он не определен, и любая версия приложения может быть использована.

8.3.3.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.3.3.3 Тип

Тип элемента xsd:string.

8.3.3.4 Атрибуты

Атрибуты не определены.

8.3.3.5 Псевдосхема

<ApplicationVersion> xsd:string </ApplicationVersion>?

8.3.3.6 Примеры

Приложение BLAST версии 1.4:

<jsdl:Application>
<jsdl:ApplicationName> BLAST </jsdl:ApplicationName>
<jsdl:ApplicationVersion> 1.4 </jsdl:ApplicationVersion>
...
</jsdl:Application>

8.4 Элементы ресурсов

Элементы ресурсов описывают ресурсные требования задачи. Может быть использовано не более одного элемента ресурсов. Как это показано в 5.1, элементы ресурсов поддерживают только элементы базового набора. Основное внимание уделяется однотипному описанию ресурсов. Дополнительные требования к ресурсам могут быть определены в качестве расширений.

8.4.1 Элемент Resources

8.4.1.1 Определение

Этот элемент содержит требования задачи к ресурсам. Если этот элемент отсутствует, потребляющая система может использовать любой набор ресурсов для выполнения задачи.

Любая комбинация перечисленных ниже подэлементов данного элемента может присутствовать в элементе Resources документа JSDL. В частности, любая комбинация элементов Individual и Total одинаковых или различных типов может иметь место в элементе Resources. Но следует заметить, что все элементы, представленные в документе JSDL, обязаны соответствовать всему документу, который должен быть удовлетворен (потребляющей системой). Могут существовать документы JSDL, которые

не могут быть удовлетворены ни при каких условиях, например в случае, когда определены противоречащие требования к ресурсам одного и того же типа Individual и Total (см. раздел 6).

8.4.1.2 Множественность

Множественность этого элемента равна либо нулю, либо единице. 8.4.1.3 Тип

Это комплексный тип. Он должен поддерживать следующие элементы:

- CandidateHosts;
- FileSystem;
- ExclusiveExecution;
- OperatingSystem;
- CPUArchitecture;
- IndividualCPUSpeed;
- IndividualCPUTime;
- IndividualCPUCount;
- IndividualNetworkBandwidth;
- IndividualPhysicalMemory;
- IndividualVirtualMemory;
- IndividualDiskSpace;
- TotalCPUTime;
- TotalCPUCount;
- TotalPhysicalMemory;
- TotalVirtualMemory;
- TotalDiskSpace;
- TotalResourceCount.

8.4.1.4 Атрибуты

Атрибуты не определены.

8.4.1.5 Псевдосхема

```
<Resources>
<CandidateHosts ... />?
<FileSystem .../>*
<ExclusiveExecution .../>?
<OperatingSystem .../>?
<CPUArchitecture .../>?
<IndividualCPUSpeed .../>?
<IndividualCPUTime .../>?
<IndividualCPUCount .../>?
<IndividualNetworkBandwidth .../>?
<IndividualPhysicalMemory .../>?
<IndividualVirtualMemory .../>?
<IndividualDiskSpace .../>?
<TotalCPUTime .../>?
<TotalCPUCount .../>?
<TotalPhysicalMemory .../>?
<TotalVirtualMemory .../>?
<TotalDiskSpace .../>?
<TotalResourceCount .../>?
<xsd:any##other>*
</Resources>?
```

8.4.2 Элемент CandidateHosts

8.4.2.1 Определение

Этот элемент является комплексным типом, который определяет набор поименованных узлов, которые могут быть выбраны для решения задачи. Если этот элемент существует, то один или более узлов из набора должен быть выбран для выполнения задачи. Если элемент отсутствует, то он не определен. Поименованный узел может быть одним узлом (т. е. представлять собой имя ЭВМ), логической группой узлов (т. е. представлять собой поименованную логическую группу или кластер), виртуальной машиной и пр.

8.4.2.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.4.2.3 Тип

Это комплексный тип. Он должен поддерживать следующие элементы:

HostName.

8.4.2.4 Атрибуты

Атрибуты не определены.

8.4.2.5 Псевдосхема

<CandidateHosts>
<HostName .../>+
</CandidateHosts>?

8.4.2.6 Примеры

Должна использоваться ЭВМ bach.example.com:

<jsdl:CandidateHosts>
<jsdl:HostName> bach.example.com </jsdl:HostName>
</jsdl:CandidateHosts>

Список машин, которые должны быть использованы:

<jsdl:CandidateHosts>
<jsdl:HostName> bach.example.com </jsdl:HostName>
<jsdl:HostName> mozart.example.com </jsdl:HostName>
<jsdl:HostName> beethoven.example.com </jsdl:HostName>
</jsdl:CandidateHosts>

Некоторые или все машины могут быть использованы.

8.4.3 Элемент HostName

8.4.3.1 Определение

Этот элемент является простым типом, содержащим единственное имя узла. Имя может соответствовать одному узлу (т. е. имени ЭВМ), логической группе узлов (т. е. имени логической группы или кластера), виртуальной машине и пр.

8.4.3.2 Множественность

Множественность этого элемента равна нулю и более.

8.4.3.3 Тип

Тип этого элемента xsd:string.

8.4.3.4 Атрибуты

Атрибуты не определены.

8.4.3.5 Псевдосхема

<HostName> xsd:string </HostName>+

8.4.3.6 Примеры

ЭВМ bach.example.com:

<jsdl:HostName> bach.example.com </jsdl:HostName>

Логическая группа с именем — the-composers:

<jsdl:HostName> the-composers </jsdl:HostName>

8.4.4 Элемент FileSystem

8.4.4.1 Определение

Данный элемент описывает файловую систему, которая требуется задачей. Это комплексный тип, который может содержать адрес, по которому файловая система была бы доступна, требуемое количество дискового пространства и тип файловой системы. Файловая система может быть локальным ресурсом (например, размещаться на локальном диске) либо быть удаленным ресурсом (например, быть смонтирована как NFS).

Данный элемент является описанием желательной конфигурации требуемых ресурсов. Он не задает действие, которое должно быть проведено на потребляющей системе. Ресурс, который не обеспечивает требуемую файловую систему, не соответствует требованиям задачи.

Отдельный механизм конфигурации может создавать ресурс с требуемой файловой системой (например, при помощи монтирования удаленной файловой системы с использованием NFS). Однако такой механизм лежит вне действия данной спецификации.

8.4.4.2 Множественность

Множественность этого элемента равна нулю и более.

8.4.4.3 Тип

Элемент является комплексным типом и должен поддерживать следующие элементы:

- FileSystemType;
- Description;
- MountPoint;
- DiskSpace.

8.4.4.4 Атрибуты

Определены следующие атрибуты:

name — имя элемента FileSystem. Атрибут имеет тип xsd:NCName. Имя определяется пользователем и должно быть уникальным в документе JSDL. Малое число хорошо известных файловых систем, о которых предполагается, что они поддерживаются везде, являются определенными и также могут использоваться в JSDL. Подробности содержатся в 8.4.4.6.

8.4.4.5 Псевдосхема

```
<FileSystem name="xsd:NCName">
<FileSystemType ... />?
<Description ... />?
<MountPoint ... />?
<DiskSpace ... />?
<xsd:any##other/>*
</FileSystem>*
```

8.4.4.6 Имена хорошо известных файловых систем

Предполагается, что потребляющая система обычно поддерживает малое число хорошо известных файловых систем. Здесь перечислены имена и семантики этих хорошо известных файловых систем совместно с минимальными нормативными определениями. Эти файловые системы недоступны по умолчанию; если они нужны для выполнения задачи, они должны быть определены в документе JSDL. Однако не существует требований, обязывающих потребляющую систему поддерживать эти файловые системы.

Для фактического представления XML декларации в дальнейшем могут быть специализированы посредством определения точки монтировки, требуемого доступного пространства файловой системы и др. Специализированные определения должны быть согласованы с минимальными нормативными определениями; FileSystemType не должен быть изменен.

Домашняя файловая система "HOME" ссылается на домашнюю директорию пользователя и может быть разделена между различными ресурсами. Тип файловой системы — "normal".

```
<jsdl:FileSystem name="HOME">
    <jsdl:FileSystemType>normal</jsdl:FileSystemType>
</jsdl:FileSystem>
```

Файловая система "ROOT" ссылается на корневой каталог, и ее использование нежелательно. Тип файловой системы — "normal". Система не разделяется между различными ресурсами. В предположении возможности записи эта файловая система оказывается незащищенной. Фактическая корневая директория файловой системы зависит от операционной системы и является непереносимой; в системах, поддерживающих POSIX, корневой каталог ссылается на "/".

```
<jsdl:FileSystem name="ROOT">
    <jsdl:FileSystemType>normal</jsdl:FileSystemType>
</jsdl:FileSystem>
```

Файловая система "SCRATCH" ссылается на временное пространство, которое сохраняется некоторое время после завершения задачи. Ее тип — "spool". Предполагается, что эта файловая система способна воспринимать относительно большие файлы.

Эта файловая система может быть разделена между различными вычислительными ресурсами.

```
<jsdl:FileSystem name="SCRATCH">
    <jsdl:FileSystemType>spool</jsdl:FileSystemType>
</jsdl:FileSystem>
```

Файловая система "TMP" ссылается на временное пространство, которое необязательно существует после окончания задачи. Тип файловой системы — "temporary". Она не разделяется между ресурсами. Предполагается, что она может быть быстродействующей.

```
<jsdl:FileSystem name="TMP">
    <jsdl:FileSystemType>temporary</jsdl:FileSystemType>
</jsdl:FileSystem>
```

8.4.4.7 Вопросы безопасности

Различные реализации систем могут принять решение не предоставлять доступ к корневой файловой системе "ROOT" по соображениям безопасности (см. также 8.4.5.6).

8.4.4.8 Примеры

Файловая система должна иметь размер не более 10 Гб. Исполняющая система может сама выбрать точку монтировки, потому что MountPoint не определен.

```
<jsdl:FileSystem name="HOME">
<jsdl:FileSystemType>normal</jsdl:FileSystemType>
<jsdl:Description> Steve's home </jsdl:Description>
<jsdl:DiskSpace>
<jsdl:UpperBoundedRange>10737418240.0</jsdl:UpperBoundedRange>
</jsdl:DiskSpace>
</jsdl:FileSystem>
```

Файловая система "HOME" должна иметь не менее 1 Гб и располагаться в месте, определенном MountPoint.

```
<jsdl:FileSystem name="HOME">
<jsdl:FileSystemType>normal</jsdl:FileSystemType>
<jsdl:Description> Ali's home </jsdl:Description>
<jsdl:MountPoint>/home/ali</jsdl:MountPoint>
<jsdl:DiskSpace>
<jsdl:LowerBoundedRange>1073741824.0</jsdl:LowerBoundedRange>
</jsdl:DiskSpace>
</jsdl:FileSystem>
```

Файловая система "TMP" должна иметь не менее 1 Гб. Исполняющая система самостоятельно выбирает подходящую точку монтировки.

```
<jsdl:FileSystem name="TMP">
<jsdl:FileSystemType>temporary</jsdl:FileSystemType>
<jsdl:DiskSpace>
<jsdl:LowerBoundedRange>1073741824.0</jsdl:LowerBoundedRange>
</jsdl:DiskSpace>
</jsdl:FileSystem>
```

Файловая система "FASTSTORAGE" должна быть доступна. Исполняющая система самостоятельно выбирает подходящую точку монтировки. Расширение описывает дополнительные требования, например высокую производительность.

```
<jsdl:FileSystem name="FASTSTORAGE">
<jsdl:FileSystemType>normal</jsdl:FileSystemType>
<jsdl:DiskSpace>
<jsdl:LowerBoundedRange>1073741824.0</jsdl:LowerBoundedRange>
</jsdl:DiskSpace>
<tns:Performance>high</tns:Performance>
</jsdl:FileSystem>
```

8.4.5 Элемент MountPoint

8.4.5.1 Определение

Этот элемент является строкой, описывающей положение точки монтировки на локальном узле, которая должна быть сделана доступной для размещения ресурсов задачи.

Если элемент не определен, потребляющая система должна выбрать локальное расположение, которое обеспечит работу с запрашиваемой файловой системой.

8.4.5.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.4.5.3 Тип

Тип элемента xsd:string.

8.4.5.4 Атрибуты

Атрибуты не определены.

8.4.5.5 Псевдосхема

<MountPoint> xsd:string </MountPoint>?

8.4.5.6 Вопросы безопасности

Не предполагается, что возможен переход вовне указанной точки монтировки, например переход в верхнюю по отношению к точке монтировки директорию каталога. Конкретная реализация потребляющей системы может запретить изменение директории выше точки монтировки по соображениям безопасности, или просто такое изменение является невозможным. Например, узел, работающий под управлением операционной системы UNIX, не может выполнить переход выше корневого каталога «/», тогда как узел, работающий под управлением операционной системы WINDOWS, не может выполнить переход выше буквенного обозначения логического диска.

Реализации потребляющей системы могут ограничить набор символов, который можно использовать для указания локального пути, и запретить использование символов "." и ".." по соображениям безопасности.

8.4.5.7 Примеры

Файловая система "HOME" должна быть сделана доступной в локальной директории "/home/darren":

```
<jsdl:FileSystem name="HOME">
...
<jsdl:MountPoint>/home/darren</jsdl:MountPoint>
</jsdl:FileSystem>
```

Файловая система "HOME" должна быть сделана доступной в локальной директории: "c:\Documents and Settings\fred":

```
<jsdl:FileSystem name="HOME">
...
<jsdl:MountPoint>c:\Documents and Settings\fred</jsdl:MountPoint>
</jsdl:FileSystem>
```

Для работы приложения, удовлетворяющего требованиям стандарта POSIX, была запрошена файловая система "HOME" без указания точки монтировки. Переменная среды "HOME" определяется посредством фактического значения точки монтировки на локальном узле.

```
<jsdl:Application>
...
<jsdl-posix:POSIXApplication>
...
<jsdl-posix:Environment name="HOME" filesystemName="HOME"/>
</jsdl-posix:POSIXApplication>
</jsdl:Application>
...
<jsdl:FileSystem name="HOME">
<jsdl:FileSystemType>normal</jsdl:FileSystemType>
...
</jsdl:FileSystem>
```

Предположим, что затребованная файловая система создана и расположена в "/usr/x00a10". Тогда среда задачи будет содержать следующую переменную среды:

HOME = /usr/x00a10

8.4.6 Элемент MountSource

Замечание:

XML-представление элемента MountSource не определено в спецификации JSDL версии 1.0. Пример использования этого элемента приведен в приложении В. Элемент jsdl:FileSystem не должен содержать более одного элемента jsdl-errata:MountSource.

8.4.6.1 Определение

Этот элемент является строкой, описывающей удаленное положение точки монтировки, которая должна быть сделана доступной с локального ресурса для выполнения задачи.

8.4.6.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.4.6.3 Тип

Тип элемента xsd:string.

8.4.6.4 Атрибуты

Атрибуты не определены.

8.4.6.5 Псевдосхема

<MountSource> xsd:string </MountSource>?

8.4.6.6 Примеры

Файловая система "HOME" должна иметь не более 10 Гб. Исполняющая система самостоятельно определяет подходящую точку монтировки, потому что не указано значение элемента MountPoint. Используя NFS, должна быть смонтирована удаленная файловая система "sputnick.ggf.org:/home/steve":

```
<jsdl:FileSystem name="HOME">
<jsdl:FileSystemType>normal</jsdl:FileSystemType>
<jsdl:Description> Steve's home </jsdl:Description>
<jsdl:DiskSpace>
<jsdl:UpperBoundedRange>10737418240.0</jsdl:UpperBoundedRange>
</jsdl:DiskSpace>
<jsdl-errata:MountSource>
nfs:sputnick.ggf.org/home/steve
</jsdl-errata:MountSource>
</jsdl:FileSystem>
```

8.4.7 Элемент DiskSpace

8.4.7.1 Определение

Этот элемент является значением диапазона, который описывает требуемое количество свободного дискового пространства в файловой системе, определенной в существующем элементе FileSystem для данной задачи. Требуемое количество свободного пространства задается в байтах. Если элемент отсутствует, то значение диапазона считается неопределенным, и потребляющая система может сама установить это значение.

8.4.7.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.4.7.3 Тип

Тип элемента jsdl:RangeValue_Type.

8.4.7.4 Атрибуты

Атрибуты не определены.

8.4.7.5 Псевдосхема

<DiskSpace> jsdl:RangeValue_Type </DiskSpace>?

8.4.7.6 Примеры

Файловая система, имеющая минимум 1 Гб свободного пространства:

```
<jsdl:FileSystem name="HOME">
...
<jsdl:DiskSpace>
<jsdl:LowerBoundedRange>1073741824.0</jsdl:LowerBoundedRange>
</jsdl:DiskSpace>
</jsdl:FileSystem>
```

Файловая система имеет максимум 1 Гб свободного пространства:

```
<jsdl:FileSystem name="TMP">
...
<jsdl:DiskSpace>
<jsdl:UpperBoundedRange>1073741824.0</jsdl:UpperBoundedRange>
</jsdl:DiskSpace>
</jsdl:FileSystem>
```

8.4.8 Элемент FileSystemType

8.4.8.1 Определение

Этот элемент является маркером, описывающим тип файловой системы, содержащейся в элементе FileSystem. Если элемент отсутствует, потребляющая система может выбрать любое значение.

8.4.8.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.4.8.3 Тип

Тип элемента jsdl:FileSystemTypeEnumeration.

8.4.8.4 Атрибуты

Атрибуты не определены.

8.4.8.5 Псевдосхема

<FileSystemType> jsdl:FileSystemTypeEnumeration </FileSystemType>?

8.4.8.6 Примеры

<jsdl:FileSystem name="HOME">
<jsdl:FileSystemType>normal</jsdl:FileSystemType>
...
</jsdl:FileSystem>

8.4.9 Элемент ExclusiveExecution

8.4.9.1 Определение

Этот элемент является логическим и обозначает, должна ли задача иметь исключительный доступ к ресурсам, предоставленным ей потребляющей системой. Если элемент отсутствует, то его значение не определено, и потребляющая система может выбрать любое значение.

8.4.9.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.4.9.3 Тип

Тип элемента xsd:boolean

- True — исключительно одна задача использует предоставленные ресурсы;
- False — другие задачи одновременно могут работать на выделенном ресурсе.

8.4.9.4 Атрибуты

Атрибуты не определены.

8.4.9.5 Псевдосхема

<ExclusiveExecution> xsd:boolean </ExclusiveExecution>?

8.4.9.6 Примеры

Выполнение исключительно одной задачи:

<jsdl:ExclusiveExecution>true</jsdl:ExclusiveExecution>

8.4.10 Элемент OperatingSystem

8.4.10.1 Определение

Этот элемент является комплексным типом, который определяет операционную систему, требуемую для выполнения задачи. Он может содержать элементы Description, OperatingSystemVersion, OperatingSystemType. Если элемент отсутствует, то его значение не определено, и потребляющая система может выбрать любое значение.

8.4.10.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.4.10.3 Тип

Элемент имеет комплексный тип. Он должен поддерживать следующие элементы:

- OperatingSystemType;
- OperatingSystemVersion;
- Description.

8.4.10.4 Атрибуты

Атрибуты не определены.

8.4.10.5 Псевдосхема

<OperatingSystem>
<OperatingSystemType ... />?
<OperatingSystemVersion ... />?
<Description ... />?
<xsd:any##other/>*
</OperatingSystem>?

8.4.11 Элемент OperatingSystemType

8.4.11.1 Определение

Этот элемент является комплексным типом, который содержит название операционной системы. Если элемент отсутствует, то его значение не определено, и потребляющая система может выбрать любое значение. Значения, не определенные в OperatingSystemTypeEnumeration (см. 7.2.3), могут быть использованы посредством определения маркера "other", и значение может быть задано в качестве расширения (см. 9.3). См. примеры, приведенные ниже.

8.4.11.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.4.11.3 Тип

Элемент имеет комплексный тип. Он должен поддерживать следующие элементы:

- OperatingSystemName.

8.4.11.4 Атрибуты

Атрибуты не определены.

8.4.11.5 Псевдосхема

```
<OperatingSystemType>
<OperatingSystemName>
<xsd:any##other/>*
</OperatingSystemType>?
```

8.4.11.6 Примеры

Операционная система Inferno:

```
<jsdl:OperatingSystemType>
<jsdl:OperatingSystemName>Inferno</jsdl:OperatingSystemName>
</jsdl:OperatingSystemType>
```

Операционная система Windows 2003 Server (используется механизм расширения JSDL "other"):

```
<jsdl:OperatingSystemType>
<jsdl:OperatingSystemName>other</jsdl:OperatingSystemName>
<tns:OtherOS>Windows_2003_Server</tns:OtherOS>
</jsdl:OperatingSystemType>
```

8.4.12 Элемент OperatingSystemName

8.4.12.1 Определение

Этот элемент является маркером, который содержит имя операционной системы.

8.4.12.2 Множественность

Множественность этого элемента равна единице.

8.4.12.3 Тип

Элемент имеет тип jsdl:OperatingSystemTypeEnumeration.

8.4.12.4 Атрибуты

Атрибуты не определены.

8.4.12.5 Псевдосхема

```
<OperatingSystemName>
jsdl:OperatingSystemTypeEnumeration
</OperatingSystemName>
```

8.4.12.6 Примеры

Операционная система Inferno:

```
<jsdl:OperatingSystemName>Inferno</jsdl:OperatingSystemName>
```

8.4.13 Элемент OperatingSystemVersion

8.4.13.1 Определение

Этот элемент является строкой, определяющей версию операционной системы, требуемой для выполнения задачи. Потребляющая система должна использовать точное текстовое совпадение для выбора версии операционной системы. Если элемент отсутствует, может быть использована любая версия операционной системы.

8.4.13.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.4.13.3 Тип

Элемент имеет тип xsd:string.

8.4.13.4 Атрибуты

Атрибуты не определены.

8.4.13.5 Псевдосхема

```
<OperatingSystemVersion> xsd:string </OperatingSystemVersion>?
```

8.4.13.6 Примеры

Операционная система, имеющая версию номер «5.01»:

```
<jsdl:OperatingSystemVersion>5.01</jsdl:OperatingSystemVersion>
```

8.4.14 Элемент CPUArchitecture

8.4.14.1 Определение

Данный элемент является строкой, определяющей архитектуру центрального процессора, требуемого для задачи в среде выполнения. Если элемент отсутствует, его значение не определено, и потребляющая система может использовать любое значение. Значения, не определенные в ProcessorArchitectureEnumeration (см. 7.2.1), могут быть использованы посредством определения

маркера "other", и значение может быть задано в качестве расширения (см. 9.3). См. примеры, приведенные ниже.

8.4.14.2 Множественность

Множественность этого элемента равна нулю либо единице.

8.4.14.3 Тип

Элемент имеет комплексный тип. Он должен поддерживать следующие элементы:

- CPUArchitectureName.

8.4.14.4 Атрибуты

Атрибуты не определены.

8.4.14.5 Псевдосхема

<CPUArchitecture>
<CPUArchitectureName .../>
<xsd:any##other>*
</CPUArchitecture>?

8.4.14.6 Примеры

ЭВМ с архитектурой SPARC:

<jsdl:CPUArchitecture>
<jsdl:CPUArchitectureName>sparc</jsdl:CPUArchitectureName>
</jsdl:CPUArchitecture>

ЭВМ с архитектурой CELL (используется механизм расширения JSDL "other"):

<jsdl:CPUArchitecture>
<jsdl:CPUArchitectureName>other</jsdl:CPUArchitectureName>
<tns:OtherCPUArchitectures>cell</tns:OtherCPUArchitectures>
</jsdl:CPUArchitecture>

8.4.15 Элемент CPUArchitectureName

8.4.15.1 Определение

Этот элемент является маркером, определяющим архитектуру центрального процессора, требуемого для задачи в среде выполнения.

8.4.15.2 Множественность

Множественность этого элемента равна единице.

8.4.15.3 Тип

Элемент имеет тип jsdl:ProcessorArchitectureEnumeration.

8.4.15.4 Атрибуты

Атрибуты не определены.

8.4.15.5 Псевдосхема

<CPUArchitectureName>
jsdl:ProcessorArchitectureEnumeration
</CPUArchitectureName>

8.4.15.6 Примеры

ЭВМ с архитектурой SPARC:

<jsdl:CPUArchitectureName>sparc</jsdl:CPUArchitectureName>

8.4.16 Элемент IndividualCPUSpeed

8.4.16.1 Определение

Этот элемент имеет значение диапазона, которое указывает быстродействие каждого центрального процессора, требуемого задачей в среде выполнения. Величина задается в герцах. Если элемент отсутствует, то он не определен, и потребляющая система может выбрать любое значение.

8.4.16.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.4.16.3 Тип

Элемент имеет тип is jsdl:RangeValue_Type.

8.4.16.4 Атрибуты

Атрибуты не определены.

8.4.16.5 Псевдосхема

<IndividualCPUSpeed> jsdl:RangeValue_Type </IndividualCPUSpeed>?

8.4.16.6 Примеры

Центральный процессор, который имеет скорость не менее 1 Гц:
<jsdl:IndividualCPUSpeed>
<jsdl:LowerBoundedRange>1073741824.0</jsdl:LowerBoundedRange>
</jsdl:IndividualCPUSpeed>

8.4.17 Элемент IndividualCPUTime

8.4.17.1 Определение

Этот элемент имеет значение диапазона, которое определяет общее число секунд работы центрального процессора, требуемого на каждом из ресурсов для выполнения задачи. Если элемент отсутствует, то он не определен, и потребляющая система может выбрать любое значение.

Если элемент используется с верхней границей, то его значение действует как максимальное число секунд работы процессора, которое задача будет использовать на индивидуальном вычислительном ресурсе. Если время выполнения задачи достигнет этого максимального значения, потребляющая система может завершить выполнение задачи.

Если элемент используется с нижней границей, то его значение представляет минимальное число секунд работы процессора, в течение которых задача гарантированно будет использовать ресурс. Потребляющая система не должна завершать выполнение задачи, пока не будет достигнута эта нижняя граница.

Предполагается, что процессорное время будет использоваться в течение достаточно непрерывного времени (т. е. непрерывно). В общем этот элемент должен рассматриваться совместно с атрибутами, применяемыми при работе расписания, которые могут быть определены в других спецификациях.

8.4.17.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.4.17.3 Тип

Элемент имеет тип is jsdl:RangeValue_Type.

8.4.17.4 Атрибуты

Атрибуты не определены.

8.4.17.5 Псевдосхема

<IndividualCPUTime> jsdl:RangeValue_Type </IndividualCPUTime>?

8.4.17.6 Примеры

Использование максимум 60 с процессорного времени:
<jsdl:IndividualCPUTime>
<jsdl:UpperBoundedRange>60.0</jsdl:UpperBoundedRange>
</jsdl:IndividualCPUTime>

8.4.18 Элемент IndividualCPUCount

8.4.18.1 Определение

Данный элемент имеет значение диапазона, которое определяет число центральных процессоров, требуемое на каждом индивидуальном ресурсе. Если элемент отсутствует, то он не определен, и потребляющая система может выбрать любое значение.

Данный элемент связан с элементом TotalCPUCount (см. 8.4.24). Если оба элемента представлены в документе, то их значения должны соответствовать друг другу, иначе документ JSDL не может быть удовлетворен (см. также 8.4.1.1).

Замечание по поводу дробных значений элемента IndividualCPUCount:

элемент IndividualCPUCount имеет тип jsdl:RangeValueType. Формально тип допускает дробные значения. Интерпретация дробных значений существенно зависит от системы, и разработчикам следует воздерживаться от использования нецелых значений элемента IndividualCPUCount по соображениям интероперабельности.

8.4.18.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.4.18.3 Тип

Элемент имеет тип is jsdl:RangeValue_Type.

8.4.18.4 Атрибуты

Атрибуты не определены.

8.4.18.5 Псевдосхема

<IndividualCPUCount> jsdl:RangeValue_Type </IndividualCPUCount>?

8.4.18.6 Примеры

Каждый из предоставленных ресурсов должен иметь два центральных процессора для задачи:

<jsdl:IndividualCPUCount>
<jsdl:Exact>2.0</jsdl:Exact>
</jsdl:IndividualCPUCount>

8.4.19 Элемент IndividualNetworkBandwidth

8.4.19.1 Определение

Этот элемент имеет значение диапазона, которое определяет требование, налагаемое на полосу пропускания сети для каждого отдельного ресурса. Значение указывают в битах в секунду. Если элемент отсутствует, то он не определен, и потребляющая система может выбрать любое значение.

Этот элемент определяет требование на максимальное (номинальное) значение полосы пропускания одного сетевого интерфейса. Ресурс, соответствующий этому требованию, должен иметь минимум один интерфейс (физический или какой-либо другой), который удовлетворяет этому требованию. Однако не существует требования о том, чтобы ресурс обязательно имел хотя бы один сетевой интерфейс или, если ресурс имеет много сетевых интерфейсов, чтобы все они удовлетворяли предъявленному требованию.

Если ресурс имеет много сетевых интерфейсов и только часть из них соответствует предъявленному требованию, должен существовать механизм, обеспечивающий предоставление задаче именно тех интерфейсов, которые удовлетворяют требованию. Описание работы такого механизма лежит вне данной спецификации.

Следует обратить внимание на то, что данный элемент предназначен только для того, чтобы выразить простейшие требования к сетевым ресурсам. Он не предназначен для описания расширенных сетевых требований, таких как требования к множественным сетевым интерфейсам, или сетевых требований, налагаемых на пару или группу ресурсов.

8.4.19.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.4.19.3 Тип

Элемент имеет тип is jsdl:RangeValue_Type.

8.4.19.4 Атрибуты

Атрибуты не определены.

8.4.19.5 Псевдосхема

<IndividualNetworkBandwidth>
jsdl:RangeValue_Type
</IndividualNetworkBandwidth>?

8.4.19.6 Примеры

Ресурс с полосой пропускания сети минимум 100 Мбит/с:

<jsdl:IndividualNetworkBandwidth>
<jsdl:LowerBoundedRange>104857600.0</jsdl:LowerBoundedRange>
</jsdl:IndividualNetworkBandwidth>

8.4.20 Элемент IndividualPhysicalMemory

8.4.20.1 Определение

Этот элемент имеет значение диапазона, которое определяет количество физической памяти, требуемой на каждом отдельном ресурсе. Величина указывается в байтах. Если элемент отсутствует, то он не определен, и потребляющая система может выбрать любое значение.

Этот элемент связан с элементом TotalPhysicalMemory (см. 8.4.25). Если оба элемента присутствуют в документе JSDL, их значения должны не противоречить друг другу, иначе не будет существовать возможности удовлетворить предъявленный документ JSDL (см. также 8.4.1.1).

Этот элемент определяет требование к количеству оперативной памяти, установленной на ресурсе, а не к количеству свободной памяти.

8.4.20.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.4.20.3 Тип

Элемент имеет тип is jsdl:RangeValue_Type.

8.4.20.4 Атрибуты

Атрибуты не определены.

8.4.20.5 Псевдосхема

<IndividualPhysicalMemory>
jsdl:RangeValue_Type
</IndividualPhysicalMemory>?

8.4.20.6 Примеры

Каждый ресурс должен иметь минимум 1 Гб физической памяти:

<jsdl:IndividualPhysicalMemory>
<jsdl:LowerBoundedRange>1073741824.0</jsdl:LowerBoundedRange>
</jsdl:IndividualPhysicalMemory>

8.4.21 Элемент IndividualVirtualMemory

8.4.21.1 Определение

Этот элемент имеет значение диапазона, которое определяет количество виртуальной памяти, требуемой на каждом отдельном ресурсе. Величина указывается в байтах. Если элемент отсутствует, то он не определен, и потребляющая система может выбрать любое значение.

Этот элемент связан с элементом TotalVirtualMemory (см. 8.4.26). Если оба элемента присутствуют в документе JSDL, их значения должны не противоречить друг другу, иначе не будет существовать возможности удовлетворить предъявленный документ JSDL (см. раздел 8.4.1.1).

Этот элемент определяет требование к количеству виртуальной памяти, сконфигурированной на ресурсе, а не к количеству свободной виртуальной памяти.

8.4.21.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.4.21.3 Тип

Элемент имеет тип is jsdl:RangeValue_Type.

8.4.21.4 Атрибуты

Атрибуты не определены.

8.4.21.5 Псевдосхема

<IndividualVirtualMemory>
jsdl:RangeValue_Type
</IndividualVirtualMemory>?

8.4.21.6 Примеры

Ресурс, имеющий минимум 1 Гб виртуальной памяти:

<jsdl:IndividualVirtualMemory>
<jsdl:LowerBoundedRange>1073741824.0</jsdl:LowerBoundedRange>
</jsdl:IndividualVirtualMemory>

8.4.22 Элемент IndividualDiskSpace

8.4.22.1 Определение

Этот элемент имеет значение диапазона, которое описывает объем дискового пространства, требуемого на каждом отдельном ресурсе. Величина указывается в байтах. Если элемент отсутствует, то он не определен, и потребляющая система может выбрать любое значение.

Этот элемент указывает требование на максимальную (в том числе и неформатированную) емкость диска отдельного ресурса. Для того чтобы ресурс соответствовал этому требованию, он должен иметь не менее одного диска, соответствующего данному требованию. Это может быть физический диск, том RAID или какой-либо другой тип блочного устройства. Однако не существует требования о том, чтобы ресурс имел хотя бы один диск или, если ресурс имеет множество дисков, чтобы каждый диск соответствовал этому требованию.

Элемент связан с элементом TotalDiskSpace (см. 8.4.27). Если оба элемента присутствуют в документе JSDL, их величины не должны противоречить друг другу, иначе не будет существовать возможности удовлетворить предъявленный документ JSDL (см. также 8.4.1.1).

8.4.22.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.4.22.3 Тип

Элемент имеет тип is jsdl:RangeValue_Type.

8.4.22.4 Атрибуты

Атрибуты не определены.

8.4.22.5 Псевдосхема

<IndividualDiskSpace> jsdl:RangeValue_Type </IndividualDiskSpace>?

8.4.22.6 Примеры

Каждый ресурс должен иметь не менее 1 Гб дискового пространства:

```
<jsdl:Resources>
...
<jsdl:IndividualDiskSpace>
<jsdl:LowerBoundedRange>1073741824.0</jsdl:LowerBoundedRange>
</jsdl:IndividualDiskSpace>
</jsdl:Resource>
```

Заметим, что в этом примере механизм, который может потребоваться для подготовки ресурса для выполнения задачи, не рассматривается.

8.4.23 Элемент TotalCPUTime

8.4.23.1 Определение

Этот элемент имеет значение диапазона, которое определяет требуемое общее число секунд работы центрального процессора для всех процессоров, используемых для выполнения задачи. Если элемент отсутствует, то он не определен, и потребляющая система может выбрать любое значение.

Если этот элемент используется с верхней границей диапазона, то его значение действует как максимальное число секунд работы центрального процессора, которое задача использует на всех отведенных ей ресурсах. Если задача достигнет указанного лимита, потребляющая система может завершить выполнение задачи.

Если этот элемент используется с нижней границей диапазона, его значение действует как минимальное число секунд работы центрального процессора, в течение которых задача гарантированно будет использовать на всех выделенных ей ресурсах. Потребляющая система не должна прерывать выполнение задачи до достижения указанного лимита времени.

Предполагается, что затребованное число секунд работы центрального процессора будет использоваться в течение достаточно непрерывного временного периода. В общем этот элемент должен рассматриваться совместно с атрибутами расписания приложений, которые могут быть определены в других спецификациях.

Этот элемент связан с элементом IndividualCPUTime (см. 8.4.17). Если оба элемента присутствуют в документе JSDL, их величины не должны противоречить друг другу, иначе не будет существовать возможности удовлетворить предъявленный документ JSDL (см. также 8.4.1.1).

8.4.23.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.4.23.3 Тип

Элемент имеет тип is jsdl:RangeValue_Type.

8.4.23.4 Атрибуты

Атрибуты не определены.

8.4.23.5 Псевдосхема

```
<TotalCPUTime> jsdl:RangeValue_Type </TotalCPUTime>?
```

8.4.23.6 Примеры

Не менее 600 с работы для всех процессоров:

```
<jsdl:TotalCPUTime>
<jsdl:UpperBoundedRange>600.0</jsdl:UpperBoundedRange>
</jsdl:TotalCPUTime>
```

8.4.24 Элемент TotalCPUCount

8.4.24.1 Определение

Этот элемент имеет значение диапазона, которое определяет требуемое общее число процессоров, необходимого для размещения задачи. Если элемент отсутствует, то он не определен, и потребляющая система может выбрать любое значение.

Этот элемент связан с элементом IndividualCPUCount (см. 8.4.18). Если оба элемента присутствуют в документе JSDL, их величины не должны противоречить друг другу, иначе не будет существовать возможности удовлетворить предъявленный документ JSDL (см. также 8.4.1.1).

Ненормативное замечание по поводу дробных значений элемента TotalCPUCount:

элемент TotalCPUCount имеет тип jsdl:RangeValueType. Формально тип допускает дробные значения. Интерпретация дробных значений существенно зависит от системы, и разработчикам следует воздерживаться от использования нецелых значений элемента TotalCPUCount по соображениям интероперабельности.

8.4.24.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.4.24.3 Тип

Элемент имеет тип is jsdl:RangeValue_Type.

8.4.24.4 Атрибуты

Атрибуты не определены.

8.4.24.5 Псевдосхема

<TotalCPUCount> jsdl:RangeValue_Type </TotalCPUCount>?

8.4.24.6 Примеры

Всего требуется два процессора:

<jsdl:TotalCPUCount><jsdl:Exact>2.0</jsdl:Exact></jsdl:TotalCPUCount>

8.4.25 Элемент TotalPhysicalMemory

8.4.25.1 Определение

Этот элемент имеет значение диапазона, которое определяет требуемый объем физической памяти для всей задачи на всех ресурсах. Величина дается в байтах. Если элемент отсутствует, то он не определен, и потребляющая система может выбрать любое значение.

Этот элемент связан с элементом IndividualPhysicalMemory (см. 8.4.20). Если оба элемента присутствуют в документе JSDL, их величины не должны противоречить друг другу, иначе не будет существовать возможности удовлетворить предъявленный документ JSDL (см. также 8.4.1.1).

Этот элемент указывает требование к общему количеству установленной физической памяти на всех выделенных ресурсах, а не к количеству свободной памяти.

8.4.25.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.4.25.3 Тип

Элемент имеет тип is jsdl:RangeValue_Type.

8.4.25.4 Атрибуты

Атрибуты не определены.

8.4.25.5 Псевдосхема

<TotalPhysicalMemory> jsdl:RangeValue_Type </TotalPhysicalMemory>?

8.4.25.6 Примеры

Всего 10 Гб физической памяти на всех выделенных ресурсах:

<jsdl:TotalPhysicalMemory>
<jsdl:LowerBoundedRange>10737418240.0</jsdl:LowerBoundedRange>
</jsdl:TotalPhysicalMemory>

8.4.26 Элемент TotalVirtualMemory

8.4.26.1 Определение

Этот элемент имеет значение диапазона, которое определяет требуемый объем виртуальной памяти для всей задачи на всех ресурсах. Величина дается в байтах. Если элемент отсутствует, то он не определен, и потребляющая система может выбрать любое значение.

Этот элемент связан с элементом IndividualVirtualMemory (см. 8.4.21). Если оба элемента присутствуют в документе JSDL, их величины не должны противоречить друг другу, иначе не будет существовать возможности удовлетворить предъявленный документ JSDL (см. также 8.4.1.1).

Этот элемент указывает требование на общее количество сконфигурированной виртуальной памяти на всех выделенных ресурсах, а не на количество свободной виртуальной памяти на выделенных ресурсах.

8.4.26.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.4.26.3 Тип

Элемент имеет тип is jsdl:RangeValue_Type.

8.4.26.4 Атрибуты

Атрибуты не определены.

8.4.26.5 Псевдосхема

<TotalVirtualMemory> jsdl:RangeValue_Type </TotalVirtualMemory>?

8.4.26.6 Примеры

Ресурс с минимум 1 Гб виртуальной памяти:

<jsdl:TotalVirtualMemory>
<jsdl:LowerBoundedRange>1073741824.0</jsdl:LowerBoundedRange>
</jsdl:TotalVirtualMemory>

8.4.27 Элемент TotalDiskSpace

8.4.27.1 Определение

Этот элемент имеет значение диапазона, которое описывает объем дискового пространства, требуемого для выполнения задачи на всех выделенных ресурсах. Величина указывается в байтах. Если элемент отсутствует, то он не определен, и потребляющая система мажет выбрать любое значение.

Этот элемент связан с элементом IndividualDiskSpace (см. 8.4.22). Если оба элемента присутствуют в документе JSDL, их величины должны не противоречить друг другу, иначе не будет существовать возможности удовлетворить предъявленный документ JSDL (см. также 8.4.1.1).

Этот элемент указывает требование на максимальную (в том числе и неформатированную) емкость дискового пространства. Для того чтобы ресурс удовлетворил этому требованию, он должен иметь не менее одного диска, соответствующего данному требованию. Однако не существует требования, чтобы дисковое пространство было обеспечено одним физическим или логическим диском.

8.4.27.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.4.27.3 Тип

Элемент имеет тип is jsdl:RangeValue_Type.

8.4.27.4 Атрибуты

Атрибуты не определены.

8.4.27.5 Псевдосхема

<TotalDiskSpace> jsdl:RangeValue_Type </TotalDiskSpace>?

8.4.27.6 Примеры

Задаче должно быть предоставлено не менее 10 Гб дискового пространства:

```
<jsdl:Resources>
...
<jsdl:TotalDiskSpace>
<jsdl:LowerBoundedRange>10737418240.0</jsdl:LowerBoundedRange>
</jsdl:TotalDiskSpace>
</jsdl:Resource>
```

Следует отметить, что в этом примере механизм, который может потребоваться для подготовки ресурсов для выполнения задачи, не рассматривается.

8.4.28 Элемент TotalResourceCount

8.4.28.1 Определение

Этот элемент имеет значение диапазона, которое задает общее количество ресурсов, требуемых для выполнения задачи. Если элемент отсутствует, то он не определен, и потребляющая система может выбрать любое значение.

8.4.28.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.4.28.3 Тип

Элемент имеет тип is jsdl:RangeValue_Type.

8.4.28.4 Атрибуты

Атрибуты не определены.

8.4.28.5 Псевдосхема

<TotalResourceCount> jsdl:RangeValue_Type </TotalResourceCount>?

8.4.28.6 Примеры

Пять ресурсов:

```
<jsdl:Resources>
...
<jsdl:TotalResourceCount>
<jsdl:Exact>5.0</jsdl:Exact>
</jsdl:TotalResourceCount>
</jsdl:Resources>
```

Пять ресурсов, каждый из которых предоставляет для задачи два процессора:

```
<jsdl:Resources>
...
<jsdl:IndividualCPUCount>
<jsdl:Exact>2.0</jsdl:Exact>
</jsdl:IndividualCPUCount>
```

<jsdl:TotalResourceCount>
<jsdl:Exact>5.0</jsdl:Exact>
</jsdl:TotalResourceCount>
...
</jsdl:Resources>

8.4.29 Дополнительные ресурсы

Существует возможность расширения JSDL для описания дополнительных ресурсов. Например, используя расширения, можно описать лицензии, поименованные ресурсы, библиотеки программ, пакеты программ, специальное оборудование.

8.5 Элементы потоков данных

8.5.1 Элемент DataStaging

8.5.1.1 Определение

Этот элемент определяет файлы, которые должны быть перемещены на исполняющий узел (входной поток), и файлы, которые должны быть взяты с исполняющего узла (выходной поток). Входные файлы перемещаются до начала выполнения задачи. Выходные файлы перемещаются после окончания работы задачи.

Если в элементах FileName или Source определена директория, то выполняется рекурсивное копирование. Если среда выполнения не поддерживает рекурсивное копирование, должно быть сформировано сообщение об ошибке. Спецификация этой ошибки, а также каким образом и когда она возникает, находится вне спецификации JSDL.

Обычно элемент DataStaging содержит элемент Source, либо элемент Target, либо оба этих элемента. Однако возможна ситуация, когда оба эти элемента отсутствуют. В этом случае не происходит никакого ввода и вывода данных. Элемент DeleteOnTermination, описанный в 8.5, может быть использован для удаления файлов в обоих случаях.

Возможна многократная выдача одного и того же выходного файла, для чего указывают одинаковое значение элемента FileName (при одинаковом значении элемента FileSystem) при многократном использовании элемента DataStaging, описывающего выходной поток.

Также возможно использовать одно и то же значение элемента FileName в различных элементах DataStaging, описывающих входной поток в один и тот же локальный файл. Результат такой операции является непредсказуемым. Такой подход считается устаревшим.

Элемент CreationFlag определяет, должен ли вносимый файл быть добавлен к уже существующему файлу или же должен переписать его. Этот элемент должен присутствовать в элементе DataStaging.

Элемент DeleteOnTermination может быть использован для удаления файла после окончания работы задачи. Если файл является выходным файлом, его удаление осуществляется после выполнения передачи данных. Файл может быть удален даже в том случае, если он не был создан в результате работы задачи. Например, предположим, что файл уже существует на исполняющем узле, а флаг CreationFlag имеет значение dontOverwrite. Этот файл будет уничтожен, если значение элемента DeleteOnTermination равно «true» и исполняемая задача имеет соответствующие права доступа.

Последовательность элементов DataStaging в документе JSDL не существенна, т. е. порядок следования этих элементов в документе JSDL не подразумевает какой бы то ни было последовательности, за исключением уже упомянутых случаев, касающихся выполнения задачи и проведения различных операций ввода/вывода.

Более сложные случаи передачи файлов (например, передача файлов, основанная на коде завершения задачи) в данной спецификации не описываются.

Права доступа и контроль доступа для файлов потоков данных должны управляться конкретной потребляющей системой. Эта проблема лежит вне данной спецификации.

Более сложные сценарии размещения задачи, чем описанные выше и касающиеся только размещения файлов потоков данных, находятся вне данной спецификации.

8.5.1.2 Множественность

Множественность этого элемента равна нулю и более.

8.5.1.3 Тип

Это комплексный тип. Он должен поддерживать следующие элементы:

- FileName;
- FilesystemName;
- CreationFlag;

- DeleteOnTermination;
- Source;
- Target.

8.5.1.4 Атрибуты

Определены следующие атрибуты:

name — опциональное имя элемента DataStaging. Тип атрибута xsd:NCName. Это имя определяется пользователем и должно быть уникальным внутри документа JSDL.

8.5.1.5 Псевдосхема

```
<DataStaging name="xsd:NCName"?>
<FileName ... />
<FilesystemName ... />?
<CreationFlag ... />
<DeleteOnTermination ... />?
<Source ... />?
<Target ... />?
<xsd:any##other/>*
</DataStaging>*
```

8.5.1.6 Примеры

Входной поток в файл:

```
<jsdl:DataStaging>
<jsdl:FileName>control.txt</jsdl:FileName>
<jsdl:CreationFlag>overwrite</jsdl:CreationFlag>
<jsdl:DeleteOnTermination>true</jsdl:DeleteOnTermination>
<jsdl:Source>
<jsdl:URI>gost.e-ecolog.ru                    >
</jsdl:Source>
</jsdl:DataStaging>
```

Входной поток в файл относительно заданной файловой системы:

```
<jsdl:FileSystem> ... </jsdl:FileSystem>
...
<jsdl:DataStaging>
<jsdl:FileName>control.txt</jsdl:FileName>
<jsdl:FilesystemName>HOME</jsdl:FilesystemName>
<jsdl:CreationFlag>overwrite</jsdl:CreationFlag>
<jsdl:DeleteOnTermination>true</jsdl:DeleteOnTermination>
<jsdl:Source>
<jsdl:URI>gost.e-ecolog.ru                    >
</jsdl:Source>
</jsdl:DataStaging>
```

Входной и выходной потоки данных могут быть определены элементами Source и Target так, чтобы один и тот же файл в начале работы использовался для входного потока, а по окончании работы — для выходного.

```
<jsdl:DataStaging>
<jsdl:FileName>state.txt</jsdl:FileName>
<jsdl:CreationFlag>overwrite</jsdl:CreationFlag>
<jsdl:DeleteOnTermination>true</jsdl:DeleteOnTermination>
<jsdl:Source>
<jsdl:URI>gost.e-ecolog.ru                  >
</jsdl:Source>
<jsdl:Target>
<jsdl:URI>gost.e-ecolog.ru                  >
</jsdl:Target>
</jsdl:DataStaging>
```

Множественные операции вывода данных могут быть описаны при использовании одного и того же значения элемента FileName (при одном и том же значении FileSystem) в различных элементах DataStaging.

```
<jsdl:DataStaging>
```

<jsdl:FileName>result.txt</jsdl:FileName>
<jsdl:CreationFlag>overwrite</jsdl:CreationFlag>
<jsdl:Target>
<jsdl:URI>gost.e-ecolog.ru ></jsdl:URI>
</jsdl:Target>
...
</jsdl:DataStaging>
...
<jsdl:DataStaging>
<jsdl:FileName>result.txt</jsdl:FileName>
<jsdl:CreationFlag>overwrite</jsdl:CreationFlag>
<jsdl:Target>
<jsdl:URI>gost.e-ecolog.ru ></jsdl:URI>
</jsdl:Target>
...
</jsdl:DataStaging>

8.5.2 Элемент FileName

8.5.2.1 Определение

Данный элемент является строкой, определяющей локальное имя файла (или директории) на исполняющем узле. Элемент FileName должен иметь относительный путь и использовать единственно допустимый разделитель '/'. Элемент FileName может быть иерархическим путем директорий в форме <директория>/<директория>/.../<имя>. При этом «<имя>» может быть как файлом, так и директорией.

8.5.2.2 Множественность

Множественность этого элемента равна единице.

8.5.2.3 Тип

Элемент имеет тип xsd:string.

8.5.2.4 Атрибуты

Атрибуты не определены.

8.5.2.5 Псевдосхема

<FileName> xsd:string </FileName>

8.5.2.6 Вопросы безопасности

Экземпляр потребляющей системы может ограничить разрешенный набор символов в локальном пути, в частности запретить использование символов '.', '..', которые потенциально могут воздействовать на безопасность локального узла.

8.5.2.7 Примеры

Имя файла:

<jsdl:DataStaging>
<jsdl:FileName>control.txt</jsdl:FileName>
...
</jsdl:DataStaging>

Иерархический путь, указывающий на файл:

<jsdl:DataStaging>
<jsdl:FileName>job1/input/control.txt</jsdl:FileName>
...
<jsdl:DataStaging>

Иерархический путь, указывающий на директорию:

<jsdl:DataStaging>
<jsdl:FileName>job1/input</jsdl:FileName>
...
</jsdl:DataStaging>

8.5.3 Элемент FilesystemName

8.5.3.1 Определение

Если этот элемент определен, то элемент FileName относится к описанию указанной файловой системы, на которое указывает имя. В этом случае должен существовать элемент FileSystem с тем же именем. Если элемент FilesystemName отсутствует, то его значение не определено. Если значение

FilesystemName не определено, то элемент FileName относится к рабочей директории задачи, которая задается потребляющей системой.

Следует отметить, что расширения JSDL могут также позволять определять рабочую директорию задачи. См., например, определение элемента WorkingDirectory в расширении «Вычисления на узлах, удовлетворяющих POSIX» (см. 10.1.7).

8.5.3.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.5.3.3 Тип

Элемент имеет тип xsd:NCName.

8.5.3.4 Атрибуты

Атрибуты не определены.

8.5.3.5 Псевдосхема

<FilesystemName> xsd:NCName </FilesystemName>?

8.5.3.6 Примеры

Обращение к файлу в указанной файловой системе:

<jsdl:FileSystem > ... </jsdl:FileSystem>

...

<jsdl:DataStaging>

...

<jsdl:FilesystemName>HOME</jsdl:FilesystemName>

...

</jsdl:DataStaging>

8.5.4 Элемент CreationFlag

8.5.4.1 Определение

Этот элемент определяет, может ли файл, создаваемый на локальной системе, переписать уже существующий файл или он будет добавлен к существующему файлу. Обычное значение этого элемента, которое предположительно обычно поддерживается, равно overwrite.

Случай, когда элементу CreationFlag присвоено значение dontOverwrite, а файл с тем же именем уже существует по указанному пути, не является ошибкой.

8.5.4.2 Множественность

Множественность этого элемента равна единице.

8.5.4.3 Тип

Элемент имеет тип jsdl: CreationFlagEnumeration.

8.5.4.4 Атрибуты

Атрибуты не определены.

8.5.4.5 Псевдосхема

<CreationFlag> jsdl:CreationFlagEnumeration </CreationFlag>

8.5.4.6 Примеры

Переписать существующий файл:

<jsdl:DataStaging>

...

<jsdl:CreationFlag>overwrite</jsdl:CreationFlag>

...

</jsdl:DataStaging>

8.5.5 Элемент DeleteOnTermination

8.5.5.1 Определение

Этот логический элемент определяет, должен ли файл быть уничтожен после окончания работы программы. Если элемент имеет значение «true», файл будет уничтожен либо после окончания работы программы, либо после того, как он будет передан (если файл является выходным). В противном случае файл остается при условии постоянства элемента FileSystem. Если элемент отсутствует, то поведение системы не определено и зависит от потребляющей системы.

8.5.5.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.5.5.3 Тип

Элемент имеет тип xsd:boolean.

- True — удалить файл после окончания задачи

- False — не удалять файл после окончания задачи

8.5.5.4 Атрибуты

Атрибуты не определены.

8.5.5.5 Псевдосхема

<DeleteOnTermination> xsd:boolean </DeleteOnTermination>?

8.5.5.6 Примеры

Удалить файл после окончания задачи:

```
<jsdl:DataStaging>
...
<jsdl:DeleteOnTermination>true</jsdl:DeleteOnTermination>
...
</jsdl:DataStaging>
```

8.5.6 Элемент Source

8.5.6.1 Определение

Этот элемент должен содержать путь к файлу или директории на удаленной системе. До начала выполнения задачи этот файл или директория должны быть доступны для чтения по пути, указанному (опционально) универсальным идентификатором ресурсов (URI). Если этот элемент не указан, то файл не может выступать в качестве входного.

8.5.6.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.5.6.3 Тип

Этот элемент имеет комплексный тип. Он должен поддерживать следующие элементы:

элемент URI — универсальный идентификатор ресурсов (URI).

8.5.6.4 Атрибуты

Атрибуты не определены.

8.5.6.5 Псевдосхема

```
<Source>
<URI ... />?
<xsd:any##other>*
</Source>?
```

8.5.6.6 Примеры

Ресурс — единичный файл:

```
<jsdl:DataStaging>
...
<jsdl:Source>
<jsdl:URI>gost.e-ecolog.ru                    >
</jsdl:Source>
...
</jsdl:DataStaging>
```

Ресурс — директория:

```
<jsdl:DataStaging>
...
<jsdl:Source>
<jsdl:URI>gost.e-ecolog.ru                    >
</jsdl:Source>
...
</jsdl:DataStaging>
```

Универсальный идентификатор ресурсов может определять любой протокол, не только протокол http:

```
<jsdl:DataStaging>
...
<jsdl:Source>
<jsdl:URI>ftp://foo.bar.com/~me/job1/input</jsdl:URI>
</jsdl:Source>
...
</jsdl:DataStaging>
```

или

```
<jsdl:DataStaging>
...
<jsdl:Source>
<jsdl:URI>rsync://foo.bar.com/~me/job1/input</jsdl:URI>
</jsdl:Source>
...
</jsdl:DataStaging>
```

8.5.7 Элемент URI

8.5.7.1 Определение

Этот элемент является универсальным идентификатором ресурсов. Он может определять путь (и протокол), который должен быть использован для входного или выходного файла.

8.5.7.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.5.7.3 Тип

Этот элемент имеет тип xsd:anyURI.

8.5.7.4 Атрибуты

Атрибуты не определены.

8.5.7.5 Псевдосхема

```
<URI> xsd:anyURI </URI>?
```

8.5.8 Элемент Target

8.5.8.1 Определение

Этот элемент содержит путь к файлу или директории на удаленной системе. Этот файл или директория должны быть доступны на запись по пути, указанному (опционально) в URI, после окончания работы программы. Если элемент отсутствует, файл или директория являются недоступными.

8.5.8.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

8.5.8.3 Тип

Этот элемент имеет комплексный тип. Он должен поддерживать следующие элементы:

элемент URI — универсальный идентификатор ресурсов (URI).

8.5.8.4 Атрибуты

Атрибуты не определены.

8.5.8.5 Псевдосхема

```
<Target>
<URI ... />?
<xsd:any##other>*
</Target>?
```

8.5.8.6 Примеры

Целевым объектом может быть одиночный файл:

```
<jsdl:DataStaging>
...
<jsdl:Target>
<jsdl:URI>gost.e-ecolog.ru                    >
</jsdl:Target>
...
</jsdl:DataStaging>
```

или директория:

```
<jsdl:DataStaging>
...
<jsdl:Target>
<jsdl:URI>gost.e-ecolog.ru                    >
</jsdl:Target>
...
</jsdl:DataStaging>
```

9 Расширения JSDL

Язык JSDL задает общую структуру для определения требований размещения задач. Эта структура может быть расширена для обеспечения лучшего соответствия специфическим потребностям. Язык JSDL обеспечивает два механизма расширения: использование атрибутов и использование элементов. В общем случае использование любого механизма расширения будет ограничивать интероперабельность, поэтому эти механизмы должны использоваться расчетливо и только в случае необходимости. Разработчикам, заинтересованным в расширении JSDL, рекомендуется вначале убедиться в том, что никакая другая группа еще не создала расширение, которое уже учитывает выдвигаемые требования.

Если элементы или атрибуты, определенные посредством расширений, существуют в документе JSDL, они должны поддерживаться таким же способом, что и обычные элементы или атрибуты JSDL.

Если элементы или атрибуты, определенные посредством расширений, существуют в документе JSDL, то они должны быть удовлетворены для документа JSDL, подлежащего удовлетворению потребляющей системой в целом.

Результат размещения документа JSDL на потребляющей системе не описывается данной спецификацией.

9.1 Расширения атрибутов

Каждый элемент JSDL допускает использование дополнительных атрибутов в требуемом количестве, причем пространство имен, обеспечивающее эти атрибуты, должно быть отлично от нормативного пространства имен, определенного в JSDL. В следующем примере, сокращенном для краткости, вводится атрибут «order», который определяет последовательность использования входных файлов:

```
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:JobDefinition xmlns="gost.e-ecolog.ru
xmlns:jsdl="gost.e-ecolog.ru
xmlns:o="gost.e-ecolog.ru                                    ">
<jsdl:JobDescription>
<jsdl:DataStaging o:order="1">
<jsdl:FileName>foo</jsdl:FileName>
<jsdl:CreationFlag>overwrite</jsdl:CreationFlag>
<jsdl:Source>
<jsdl:URI>gost.e-ecolog.ru                                   >
</jsdl:Source>
</jsdl:DataStaging>
<jsdl:DataStaging o:order="2">
<jsdl:FileName>bar</jsdl:FileName>
<jsdl:CreationFlag>overwrite</jsdl:CreationFlag>
<jsdl:Source>
<jsdl:URI>http://www.nowhere.com/bar-file</jsdl:URI>
</jsdl:Source>
</jsdl:DataStaging>
</jsdl:JobDescription>
</jsdl:JobDefinition>
```

9.2 Расширения элементов

В случае необходимости в рамках общей структуры документа JSDL-схема допускает использование дополнительных элементов, которые не определены в JSDL. Как и в случае расширения атрибутов, эти элементы должны быть описаны в пространстве имен, отличном от нормативного пространства имен, определенного в JSDL. В следующем примере, сокращенном для краткости, язык JSDL расширяется путем введения элемента, описывающего резервирование ресурсов:

```
...
<jsdl:Resources>
<jsdl:TotalCPUCount>
<jsdl:Exact>1.0</jsdl:Exact>
</jsdl:TotalCPUCount>
<jsdl:TotalDiskSpace>
```

```
<!-- At least 1 gigabyte disk space -->
<jsdl:LowerBoundedRange>1073741824.0</jsdl:LowerBoundedRange>
</jsdl:TotalDiskSpace>
<res:Reservation xmlns:res="gost.e-ecolog.ru                    ">
<res:Ticket>h933fsolenri900wnmd90mm34</res:Ticket>
</res:Reservation>
</jsdl:Resources>
...
```

9.3 Семантика значения «other» языка JSDL

JSDL определяет перечисляемые значения, соответствующие архитектурам процессоров и операционным системам. Оба перечисления не являются полностью завершенными. Оба перечисления внедрены в контейнерные типы, допускающие расширения, "jsdl:CPUArchitecture_Type" и "jsdl:OperatingSystemType_Type" соответственно.

Для расширения перечислений, соответствующих элементам "jsdl:CPUArchitectureName" и "jsdl:OperatingSystemName", должно использоваться специальное ключевое слово «other». При этом должен использоваться способ расширения элементов, описанный в 9.2.

Например, предположим, что существует процессор, специализирующийся на выполнении кода Java, который имеет имя JavaCPU. Этот процессор может быть описан следующим образом в секции «Ресурсы» документа JSDL:

```
...
<jsdl:Resources>
<jsdl:CPUArchitecture>
<jsdl:CPUArchitectureName>other</jsdl:CPUArchitectureName>
<ex:OtherCPUTypes xmlns:ex="gost.e-ecolog.ru                    >
JavaCPU
</ex:OtherCPUTypes>
</jsdl:CPUArchitecture>
</jsdl:Resources>
...
```

10 Нормативные расширения JSDL

Следующее расширение является нормативно определенным в JSDL 1.0.

10.1 Выполнение задач на POSIX-совместимых узлах

Это нормативное расширение определяет схему, которая описывает приложения, исполняемые на узлах, удовлетворяющих требованиям POSIX.

jsdl-posix — префикс пространства имен, используемый для данной схемы в данной спецификации. Нормативное пространство имен для этой схемы приведено в таблице 4.1.

10.1.1 Элемент POSIXApplication

10.1.1.1 Определение

Этот элемент описывает приложения, созданные в соответствии с рекомендациями POSIX. Он содержит элементы Executable, Argument, Input, Output, Error, WorkingDirectory, Environment, различные элементы, описывающие ограничения в POSIX, а также имена пользователя и группы. Если элемент существует как подэлемент элемента Application языка JSDL, то он должен появляться только один раз.

10.1.1.2 Множественность

Множественность этого элемента равна нулю либо единице.

10.1.1.3 Тип

Этот элемент имеет комплексный тип. Он должен поддерживать следующие элементы:

- Executable;
- Argument;
- Input;
- Output;
- Error;

- WorkingDirectory;
- Environment;
- WallTimeLimit;
- FileSizeLimit;
- CoreDumpLimit;
- DataSegmentLimit;
- LockedMemoryLimit;
- MemoryLimit;
- OpenDescriptorsLimit;
- PipeSizeLimit;
- StackSizeLimit;
- CPUTimeLimit;
- ProcessCountLimit;
- VirtualMemoryLimit;
- ThreadCountLimit;
- UserName;
- GroupName.

10.1.1.4 Атрибуты

Определены следующие атрибуты:

name — опциональное имя элемента POSIXApplication. Его тип xsd:NCName. Он может быть повторно использован, и ссылаться на этот атрибут можно извне документа JSDL.

10.1.1.5 Псевдосхема

```
<POSIXApplication name="xsd:NCName"?>
<Executable ... />?
<Argument ... />*
<Input ... />?
<Output ... />?
<Error ... />?
<WorkingDirectory ... />?
<Environment ... />*
<WallTimeLimit ... />?
<FileSizeLimit ... />?
<CoreDumpLimit ... />?
<DataSegmentLimit ... />?
<LockedMemoryLimit ... />?
<MemoryLimit ... />?
<OpenDescriptorsLimit ... />?
<PipeSizeLimit ... />?
<StackSizeLimit ... />?
<CPUTimeLimit ... />?
<ProcessCountLimit ... />?
<VirtualMemoryLimit ... />?
<ThreadCountLimit ... />?
<UserName ... />?
<GroupName ... />?
</POSIXApplication>?
```

10.1.2 Элемент Executable

10.1.2.1 Определение

Этот элемент является строкой, описывающей исполняемую команду. Если элемент ApplicationName не указан и элемент Executable также не указан, документ JSDL определяет пустую (нулевую) задачу или нулевой поток данных. Если оба элемента, ApplicationName и Executable, определены, элемент Executable должен определять исполняемое приложение, используя указанное имя приложения и его версию, согласно элементу ApplicationVersion, если он содержится в документе (см. также 8.3.1 и 8.3.2).

10.1.2.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

10.1.2.3 Тип

Этот элемент имеет тип xsd:string.

10.1.2.4 Атрибуты

Определены следующие атрибуты:

filesystemName — имя файловой системы, определенной элементом FileSystem внутри документа JSDL. Если атрибут присутствует, то содержательная строка элемента Executable должна интерпретироваться как имя файла, сформированного относительно точки монтировки указанной файловой системы.

10.1.2.5 Псевдосхема

<Executable filesystemName="xsd:NCName"?> xsd:string </Executable>?

10.1.2.6 Соображения безопасности

Экземпляр потребляющей системы может ограничить набор допустимых символов в локальных путях, в частности запретить использование '.' и '..', которые потенциально могут воздействовать на безопасность локального узла.

10.1.2.7 Примеры

Детально указанное исполняемое приложение:

<jsdl-posix:Executable> /usr/local/bin/gnuplot </jsdl-posix:Executable>

Использование пути, определенного в потребляющей системе:

<jsdl-posix:Executable> gnuplot </jsdl-posix:Executable>

Выполняемое приложение расположено по указанному пути относительно файловой системы "HOME":

<jsdl-posix:Executable filesystemName="HOME">
scripts/myExample
</jsdl-posix:Executable>

10.1.3 Элемент Argument

10.1.3.1 Определение

Этот элемент является ограниченной нормализованной строкой, которая указывает аргументы приложения. Элементы Argument могут быть пустыми, но не должны быть комбинированными.

Элемент Argument должен передаваться приложению как один аргумент. Пустой элемент должен передаваться в приложение пустым аргументом. Порядок, в котором аргументы передаются приложению, должен совпадать с порядком следования элементов Argument в документе.

Замечание: в будущих версиях спецификации JSDL тип этого элемента может быть изменен на xsd:string.

10.1.3.2 Множественность

Множественность этого элемента равна нулю и более.

10.1.3.3 Тип

Этот элемент имеет тип xsd:normalizedString.

10.1.3.4 Атрибуты

Определены следующие атрибуты:

filesystemName — имя файловой системы, определенной элементом FileSystem внутри документа JSDL. Если атрибут присутствует, то содержательная строка элемента Executable должна интерпретироваться как имя файла, сформированного относительно точки монтировки указанной файловой системы.

10.1.3.5 Псевдосхема

<Argument filesystemName="xsd:NCName"?>
xsd:normalizedString
</Argument>*

10.1.3.6 Соображения безопасности

Экземпляр потребляющей системы может ограничить набор допустимых символов в локальных путях, в частности запретить использование '.' и '..', которые потенциально могут воздействовать на безопасность локального узла.

10.1.3.7 Примеры

Определить CLASSPATH и класс для включения в Java-приложение:

<jsdl-posix:Argument>-cp</jsdl-posix:Argument>
<jsdl-posix:Argument>./example.jar</jsdl-posix:Argument>
<jsdl-posix:Argument>org.example.Main</jsdl-posix:Argument>

В результате будет проведена трансляция в "[java]-cp./example.jar org.example.Main". Заметим, что порядок следования аргументов должен быть сохранен. Изменение порядка аргументов, например на "[java] ./example.jar–cp org.example.Mail", очевидно, некорректно.

Запустить сервис Apache2 в окне Windows, не являясь администратором:

```
<jsdl-posix:Argument>
/user:Administrator@WORKGROUP</jsdl-posix:Argument>
<jsdl-posix:Argument>net start Apache2</jsdl-posix:Argument>
```

Выражение будет транслировано в "[runas] /user:Administrator@WORKGROUP «net start Apache2».

Отобразить в качестве эха последовательный список аргументов на стандартный вывод:

```
<jsdl-posix:Argument>foo</jsdl-posix:Argument>
<jsdl-posix:Argument>bar</jsdl-posix:Argument>
<jsdl-posix:Argument>baz</jsdl-posix:Argument>
```

Выражение будет транслировано в "[echo] foo bar baz",

```
<jsdl-posix:Argument>foo</jsdl-posix:Argument>
<jsdl-posix:Argument>bar</jsdl-posix:Argument>
<jsdl-posix:Argument></jsdl-posix:Argument>
<jsdl-posix:Argument>baz</jsdl-posix:Argument>
```

будет транслировано в "[echo] foo bar "baz",

```
<jsdl-posix:Argument>foo</jsdl-posix:Argument>
<jsdl-posix:Argument>bar</jsdl-posix:Argument>
<jsdl-posix:Argument>baz</jsdl-posix:Argument>
<jsdl-posix:Argument></jsdl-posix:Argument>
```

будет транслировано в "[echo] foo bar baz".

Отобразить в качестве «эха» путь к домашней директории пользователя на стандартный вывод:

```
<jsdl-posix:Argument filesystemName="HOME"/>
```

будет транслировано в [echo] /home/darrenp. Предполагается, что файловая система "HOME" смонтирована в директории /home/darrenp.

10.1.4 Элемент Input

10.1.4.1 Определение

Этот элемент является строкой, которая задает вход (стандартный вход) для команд. Элемент Input является именем файла, путь к которому указывается относительно либо рабочей директории, либо указанной файловой системы. Если этот элемент отсутствует, его значение не определено, и потребляющая система может выбрать произвольное значение.

10.1.4.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

10.1.4.3 Тип

Этот элемент имеет тип xsd:string.

10.1.4.4 Атрибуты

Определены следующие атрибуты:

filesystemName — имя файловой системы, определенной элементом FileSystem внутри документа JSDL. Если атрибут присутствует, то содержательная строка элемента Executable должна интерпретироваться как имя файла, сформированного относительно точки монтировки указанной файловой системы.

10.1.4.5 Псевдосхема

```
<Input filesystemName="xsd:NCName"?> xsd:string </Input>?
```

10.1.4.6 Соображения безопасности

Экземпляр потребляющей системы может ограничить набор допустимых символов в локальных путях, в частности запретить использование '.' и '..', которые потенциально могут воздействовать на безопасность локального узла.

10.1.4.7 Примеры

Стандартный ввод будет взят из файла "~/input.txt":

```
...
<jsdl-posix:Input filesystemName="HOME">
input.txt
</jsdl-posix:Input>
...
```

10.1.5 Элемент Output

10.1.5.1 Определение

Этот элемент является строкой, которая задает устройство стандартного вывода для команд. Элемент Output является именем файла, путь к которому указывается относительно либо рабочей директории, либо указанной файловой системы. Если этот элемент отсутствует, его значение не определено, и потребляющая система может выбрать произвольное значение.

10.1.5.2 Множественность

Множественность этого элемента равна либо нулю, либо единице.

10.1.5.3 Тип

Этот элемент имеет тип xsd:string.

10.1.5.4 Атрибуты

Определены следующие атрибуты:

filesystemName — имя файловой системы, определенной элементом FileSystem внутри документа JSDL. Если атрибут присутствует, то содержательная строка элемента Executable должна интерпретироваться как имя файла, сформированного относительно точки монтировки указанной файловой системы.

10.1.5.5 Псевдосхема

<Output filesystemName="xsd:NCName"?> xsd:string </Output>?

10.1.5.6 Соображения безопасности

Экземпляр потребляющей системы может ограничить набор допустимых символов в локальных путях, в частности запретить использование '.' и '..', которые потенциально могут воздействовать на безопасность локального узла.

10.1.5.7 Примеры

Стандартный вывод будет осуществляться в файл "~/output.txt":

```
...
<jsdl-posix:Output filesystemName="HOME">
output.txt
</jsdl-posix:Output>
...
```

10.1.6 Элемент Error

10.1.6.1 Определение

Этот элемент является строкой, которая задает вывод ошибок (стандартный вывод ошибок) для команд. Элемент Error является именем файла, путь к которому указывается относительно либо рабочей директории, либо указанной файловой системы. Если этот элемент отсутствует, его значение не определено, и потребляющая система может выбрать произвольное значение.

10.1.6.2 Множественность

Множественность этого элемента равна нулю либо единице.

10.1.6.3 Тип

Этот элемент имеет тип xsd:string.

10.1.6.4 Атрибуты

Определены следующие атрибуты:

filesystemName — имя файловой системы, определенной элементом FileSystem внутри документа JSDL. Если атрибут присутствует, то содержательная строка элемента Executable должна интерпретироваться как имя файла, сформированного относительно точки монтировки указанной файловой системы.

10.1.6.5 Псевдосхема

<Error filesystemName="xsd:NCName"?> xsd:string </Error>?

10.1.6.6 Соображения безопасности

Экземпляр потребляющей системы может ограничить набор допустимых символов в локальных путях, в частности запретить использование '.' и '..', которые потенциально могут воздействовать на безопасность локального узла.

10.1.6.7 Примеры

Стандартный вывод ошибок будет осуществляться в файл "~/error.txt":

```
...
<jsdl-posix:Error filesystemName="HOME">
error.txt
</jsdl-posix:Error>
...
```

10.1.7 Элемент WorkingDirectory

10.1.7.1 Определение

Этот элемент является строкой, которая задает начальную директорию выполнения задачи. Если этот элемент отсутствует, его значение не определено, и потребляющая система может выбрать произвольное значение. Во многих случаях рабочая директория может быть связана с элементом FileSystem путем задания одинакового значения элемента WorkingDirectory и локального пути монтировки элемента FileSystem. Элемент WorkingDirectory может не иметь требуемого прямого отношения к конкретному элементу FileSystem.

10.1.7.2 Множественность

Множественность этого элемента равна нулю либо единице.

10.1.7.3 Тип

Этот элемент имеет тип xsd:string.

10.1.7.4 Атрибуты

Определены следующие атрибуты:

filesystemName — имя файловой системы, определенной элементом FileSystem внутри документа JSDL. Если атрибут присутствует, то содержательная строка элемента Executable должна интерпретироваться как имя файла, сформированного относительно точки монтировки указанной файловой системы.

10.1.7.5 Псевдосхема

<WorkingDirectory filesystemName="xsd:NCName"?>
xsd:string
</WorkingDirectory>?

10.1.7.6 Соображения безопасности

Экземпляр потребляющей системы может ограничить набор допустимых символов в локальных путях, в частности запретить использование '.' и '..', которые потенциально могут воздействовать на безопасность локального узла.

10.1.7.7 Примеры

Рабочая директория задачи должна быть домашней директорией пользователя:

<WorkingDirectory filesystemName="HOME"/>

10.1.8 Элемент Environment

10.1.8.1 Определение

Этот элемент определяет имя и литерное значение переменной среды, которая будет определена для задачи в среде выполнения. Порядок, в котором переменная среды устанавливается в среде выполнения, должен соответствовать порядку следования элементов Environment в документе.

В качестве специального случая переменная среды может содержать точку монтировки файловой системы, определенной в документе JSDL. Связывание имен точки монтировки выполняется с использованием атрибута filesystemName элемента Environment и имени, содержащегося в элементе FileSystem (*см.* раздел 8.4.22).

Замечание: могут существовать потребляющие системы, которые интерпретируют элемент Environment как нелитеральный и выполняют подстановки значений. Например, предположим, что элемент Environment имеет значение "${HOME}". Потребляющая система, формируя среду выполнения задачи, должна подставить вместо этого значения строку, содержащую путь к домашней директории пользователя. Однако такое поведение является системно-зависимым (нарушает интероперабельность), его описание лежит вне спецификации JSDL.

10.1.8.2 Множественность

Множественность этого элемента равна нулю и более.

10.1.8.3 Тип

Этот элемент имеет тип xsd:string. Это значение переменной среды.

10.1.8.4 Атрибуты

Определены следующие атрибуты:

- name — имя переменной среды;
- filesystemName — имя файловой системы, определенной элементом FileSystem внутри документа JSDL. Если атрибут присутствует, то содержательная строка элемента Executable должна интерпретироваться как имя файла, сформированного относительно точки монтировки указанной файловой системы.

10.1.8.5 Псевдосхема

<Environment name="xsd:NCName" filesystemName="xsd:NCName"?>
xsd:string
</Environment>*

10.1.8.6 Соображения безопасности

Экземпляр потребляющей системы может ограничить набор допустимых символов в локальных путях, в частности запретить использование '.' и '..', которые потенциально могут воздействовать на безопасность локального узла.

10.1.8.7 Примеры

Указать, что программа оболочки должна быть bash:

<Environment name="SHELL">/bin/bash</Environment>

Установить положение файла почтового ящика на "~/mailboxes/INBOX"

<jsdl-posix:Environment name="MAIL" filesystemName="HOME">
mailboxes/INBOX
</jsdl-posix:Environment>

Установить значение переменной среды "TMPDIR" на корневой каталог файловой системы с именем "TMP":

<jsdl:FileSystem name="TMP">...</jsdl:FileSystem>
...
<jsdl-posix:Environment name="TMPDIR" filesystemName="TMP"/>
...

10.1.9 Элемент WallTimeLimit

10.1.9.1 Определение

Этот элемент является положительным целым числом, которое задает ограничение на время выполнения приложения в секундах. Если этот элемент отсутствует, потребляющая система может выбрать величину, соответствующую значению по умолчанию.

10.1.9.2 Множественность

Множественность этого элемента равна нулю либо единице.

10.1.9.3 Тип

Этот элемент имеет тип xsd:nonNegativeInteger.

10.1.9.4 Атрибуты

Атрибуты не определены.

10.1.9.5 Псевдосхема

<WallTimeLimit> xsd:nonNegativeInteger </WallTimeLimit>?

10.1.9.6 Примеры

Установить длительность выполнения равной 1 мин:

<jsdl-posix:WallTimeLimit> 60 </jsdl-posix:WallTimeLimit>

10.1.10 Элемент FileSizeLimit

10.1.10.1 Определение

Этот элемент является положительным целым числом, которое задает максимальный размер любого данного файла, связанного с описываемой задачей. Размер файла задается в байтах. Если этот элемент отсутствует, потребляющая система может выбрать значение по умолчанию.

10.1.10.2 Множественность

Множественность этого элемента равна нулю либо единице.

10.1.10.3 Тип

Этот элемент имеет тип xsd:nonNegativeInteger.

10.1.10.4 Атрибуты

Атрибуты не определены.

10.1.10.5 Псевдосхема

<FileSizeLimit> xsd:nonNegativeInteger </FileSizeLimit>?

10.1.10.6 Примеры

Установить максимальный размер файла, равный 1 Гб:

<jsdl-posix:FileSizeLimit> 1073741824 </jsdl-posix:FileSizeLimit>

10.1.11 Элемент CoreDumpLimit

10.1.11.1 Определение

Этот элемент является положительным целым числом, которое задает максимальный размер дампов (образов) памяти, которые может создать задача. Размер задается в байтах. Если этот элемент отсутствует, потребляющая система может выбрать значение по умолчанию.

10.1.11.2 Множественность

Множественность этого элемента равна нулю либо единице.

10.1.11.3 Тип

Этот элемент имеет тип xsd:nonNegativeInteger.

10.1.11.4 Атрибуты

Атрибуты не определены.

10.1.11.5 Псевдосхема

<CoreDumpLimit> xsd:nonNegativeInteger </CoreDumpLimit>?

10.1.11.6 Примеры

Запретить создание образов памяти путем задания максимально разрешенного размера, равного нулю:

<jsdl-posix:CoreDumpLimit> 0 </jsdl-posix:CoreDumpLimit>

10.1.12 Элемент DataSegmentLimit

10.1.12.1 Определение

Этот элемент является положительным целым числом, которое ограничивает размер сегмента данных. Размер дается в байтах. Если этот элемент отсутствует, потребляющая система может выбрать значение по умолчанию.

10.1.12.2 Множественность

Множественность этого элемента равна нулю либо единице.

10.1.12.3 Тип

Этот элемент имеет тип xsd:nonNegativeInteger.

10.1.12.4 Атрибуты

Атрибуты не определены.

10.1.12.5 Псевдосхема

<DataSegmentLimit> xsd:nonNegativeInteger </DataSegmentLimit>?

10.1.12.6 Примеры

Ограничить размер сегмента данных 32 Кб:

<jsdl-posix:DataSegmentLimit> 32768 </jsdl-posix:DataSegmentLimit>

10.1.13 Элемент LockedMemoryLimit

10.1.13.1 Определение

Этот элемент является положительным целым числом, которое описывает максимальный размер физической памяти, который может блокировать задача. Размер дается в байтах. Если этот элемент отсутствует, потребляющая система может выбрать значение по умолчанию.

10.1.13.2 Множественность

Множественность этого элемента равна нулю либо единице.

10.1.13.3 Тип

Этот элемент имеет тип xsd:nonNegativeInteger.

10.1.13.4 Атрибуты

Атрибуты не определены.

10.1.13.5 Псевдосхема

<LockedMemoryLimit> xsd:nonNegativeInteger </LockedMemoryLimit>?

10.1.13.6 Примеры

Разрешить блокировать не более 8 Мб памяти:

<jsdl-posix:LockedMemoryLimit> 8388608 </jsdl-posix:LockedMemoryLimit>

10.1.14 Элемент MemoryLimit

10.1.14.1 Определение

Этот элемент является положительным целым числом, которое описывает максимальный размер физической памяти, который задача будет использовать во время выполнения. Размер дается в байтах. Если этот элемент отсутствует, потребляющая система может выбрать значение по умолчанию.

10.1.14.2 Множественность

Множественность этого элемента равна нулю либо единице.

10.1.14.3 Тип

Этот элемент имеет тип xsd:nonNegativeInteger.

10.1.14.4 Атрибуты

Атрибуты не определены.

10.1.14.5 Псевдосхема

<MemoryLimit> xsd:nonNegativeInteger </MemoryLimit>?

10.1.14.6 Примеры

Установить лимит используемой физической памяти, равный 64 Мб:

<jsdl-posix:MemoryLimit> 67108864 </jsdl-posix:MemoryLimit>

10.1.15 Элемент OpenDescriptorsLimit

10.1.15.1 Определение

Этот элемент является положительным целым числом, которое описывает максимальное число открытых дескрипторов файлов (файлов, каналов, сокетов), которое может иметь задача. Если этот элемент отсутствует, потребляющая система может выбрать значение по умолчанию.

10.1.15.2 Множественность

Множественность этого элемента равна нулю либо единице.

10.1.15.3 Тип

Этот элемент имеет тип xsd:nonNegativeInteger.

10.1.15.4 Атрибуты

Атрибуты не определены.

10.1.15.5 Псевдосхема

<OpenDescriptorsLimit> xsd:nonNegativeInteger </OpenDescriptorsLimit>?

10.1.15.6 Примеры

Максимальное число дескрипторов должно быть 16:

<jsdl-posix:OpenDescriptorsLimit> 16 </jsdl-posix:OpenDescriptorsLimit>

10.1.16 Элемент PipeSizeLimit

10.1.16.1 Определение

Этот элемент является положительным целым числом, которое описывает максимальный размер каналов, созданных во время выполнения задачи. Размер задается в байтах. Если этот элемент отсутствует, потребляющая система может выбрать значение по умолчанию.

10.1.16.2 Множественность

Множественность этого элемента равна нулю либо единице.

10.1.16.3 Тип

Этот элемент имеет тип xsd:nonNegativeInteger.

10.1.16.4 Атрибуты

Атрибуты не определены.

10.1.16.5 Псевдосхема

<PipeSizeLimit> xsd:nonNegativeInteger </PipeSizeLimit>?

10.1.16.6 Примеры

Установить лимит в 512 б:

<jsdl-posix:PipeSizeLimit> 512 </jsdl-posix:PipeSizeLimit>

10.1.17 Элемент StackSizeLimit

10.1.17.1 Определение

Этот элемент является положительным целым числом, которое описывает максимальный размер стека для данной задачи. Размер задается в байтах. Если этот элемент отсутствует, потребляющая система может выбрать значение по умолчанию.

10.1.17.2 Множественность

Множественность этого элемента равна нулю либо единице.

10.1.17.3 Тип

Этот элемент имеет тип xsd:nonNegativeInteger.

10.1.17.4 Атрибуты

Атрибуты не определены.

10.1.17.5 Псевдосхема

<StackSizeLimit> xsd:nonNegativeInteger </StackSizeLimit>?

10.1.17.6 Примеры

Установить размер стека, равный 1 Мб:

<jsdl-posix:StackSizeLimit> 1048576 </jsdl-posix:StackSizeLimit>

10.1.18 Элемент CPUTimeLimit

10.1.18.1 Определение

Этот элемент является положительным целым числом, которое описывает число секунд времени центрального процессора, предназначенного для использования задачей до тех пор, пока задаче не будет передан сигнал SIGXCPU. Время задается в секундах. Если этот элемент отсутствует, потребляющая система может выбрать значение по умолчанию.

10.1.18.2 Множественность

Множественность этого элемента равна нулю либо единице.

10.1.18.3 Тип

Этот элемент имеет тип xsd:nonNegativeInteger.

10.1.18.4 Атрибуты

Атрибуты не определены.

10.1.18.5 Псевдосхема

<CPUTimeLimit> xsd:nonNegativeInteger </CPUTimeLimit>?

10.1.18.6 Примеры

Разрешить использование 30 с центрального процессора:

<jsdl-posix:CPUTimeLimit> 30 </jsdl-posix:CPUTimeLimit>

10.1.19 Элемент ProcessCountLimit

10.1.19.1 Определение

Этот элемент является положительным целым числом, которое описывает максимальное разрешенное число процессов, которое может порождать задача. Если этот элемент отсутствует, потребляющая система может выбрать значение по умолчанию.

10.1.19.2 Множественность

Множественность этого элемента равна нулю либо единице.

10.1.19.3 Тип

Этот элемент имеет тип xsd:nonNegativeInteger.

10.1.19.4 Атрибуты

Атрибуты не определены.

10.1.19.5 Псевдосхема

<ProcessCountLimit> xsd:nonNegativeInteger </ProcessCountLimit>?

10.1.19.6 Примеры

Разрешено не более восьми процессов:

<jsdl-posix:ProcessCountLimit> 8 </jsdl-posix:ProcessCountLimit>

10.1.20 Элемент VirtualMemoryLimit

10.1.20.1 Определение

Этот элемент является положительным целым числом, которое описывает максимальный разрешенный объем виртуальной памяти, который может использовать задача. Объем задается в байтах. Если этот элемент отсутствует, потребляющая система может выбрать значение по умолчанию.

10.1.20.2 Множественность

Множественность этого элемента равна нулю либо единице.

10.1.20.3 Тип

Этот элемент имеет тип xsd:nonNegativeInteger.

10.1.20.4 Атрибуты

Атрибуты не определены.

10.1.20.5 Псевдосхема

<VirtualMemoryLimit> xsd:nonNegativeInteger </VirtualMemoryLimit>?

10.1.20.6 Примеры

Максимальный предел виртуальной памяти должен быть 128 Мб:

<jsdl-posix:VirtualMemoryLimit>
134217728
</jsdl-posix:VirtualMemoryLimit>

10.1.21 Элемент ThreadCountLimit

10.1.21.1 Определение

Этот элемент является положительным целым числом, которое описывает количество потоков, которое может порождать задача. Если этот элемент отсутствует, потребляющая система может выбрать величину, соответствующую значению по умолчанию.

10.1.21.2 Множественность

Множественность этого элемента равна нулю либо единице.

10.1.21.3 Тип

Этот элемент имеет тип xsd:nonNegativeInteger.

10.1.21.4 Атрибуты

Атрибуты не определены.

10.1.21.5 Псевдосхема

<ThreadCountLimit> xsd:nonNegativeInteger </ThreadCountLimit>?

10.1.21.6 Примеры

Не более восьми потоков:

<jsdl-posix:ThreadCountLimit> 8 </jsdl-posix:ThreadCountLimit>

10.1.22 Элемент UserName

10.1.22.1 Определение

Этот элемент является строкой, которая определяет имя пользователя, используемое во время исполнения приложения. Если этот элемент отсутствует, потребляющая система может выбрать имя пользователя, базируясь на конкретном экземпляре потребляющей системы.

10.1.22.2 Множественность

Множественность этого элемента равна нулю либо единице.

10.1.22.3 Тип

Этот элемент имеет тип xsd:string.

10.1.22.4 Атрибуты

Атрибуты не определены.

10.1.22.5 Псевдосхема

<UserName> xsd:string </UserName>?

10.1.22.6 Примеры

Имя пользователя в UNIX:

<jsdl-posix:UserName>frank</jsdl-posix:UserName>

10.1.23 Элемент GroupName

10.1.23.1 Определение

Этот элемент является строкой, которая определяет имя группы, используемое во время исполнения приложения. Если этот элемент отсутствует, потребляющая система может выбрать имя группы, базируясь на конкретном экземпляре потребляющей системы.

10.1.23.2 Множественность

Множественность этого элемента равна нулю либо единице.

10.1.23.3 Тип

Этот элемент имеет тип xsd:string.

10.1.23.4 Атрибуты

Атрибуты не определены.

10.1.23.5 Псевдосхема

<GroupName> xsd:string </GroupName>?

10.1.23.6 Примеры

Имя группы в UNIX:

<jsdl-posix:GroupName>staff</jsdl-posix:GroupName>

11 Вопросы безопасности

Данная спецификация определяет язык описания требований вычислительной задачи при ее представлении (размещении) в Грид-системе или какой-либо другой системе управления выполнением задач. Предполагается, что размещение задачи должно быть безопасным, но механизм обеспечения безопасности лежит вне данной спецификации.

Возможность описания прав, необходимых для выполнения задачи, является очень важной. Ожидается, что документы JSDL версии 1.0 будут объединены с более специализированными языками описания политик и безопасности для четкого делегирования прав, если требуется.

Другие вопросы безопасности, которые являются специфическими для определенных в данной спецификации элементов, обсуждаются подробно в каждом разделе, в котором описаны эти элементы.

Приложение А
(обязательное)

Нормативная схема JSDL

Данный раздел содержит полное определение XML-схемы для JSDL 1.0.

Для удобства информативная копия этой схемы доступна онлайн по адресу gost.e-ecolog.ru . Если тем не менее будут встречаться отличия между определениями, приведенными в этом разделе, и определениями или их частями, приведенными в других разделах или в онлайн-схеме, нормативными считаются определения, приведенные в настоящем разделе.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
-->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="gost.e-ecolog.ru
xmlns:jsdl="gost.e-ecolog.ru
targetNamespace="http://schemas.ggf.org/jsdl/2005/11/jsdl"
elementFormDefault="qualified">
<!--============================================================-->
<!-- SIMPLE TYPES -->
<xsd:simpleType name="ProcessorArchitectureEnumeration">
<xsd:restriction base="xsd:string">
<xsd:enumeration value="sparc"/>
<xsd:enumeration value="powerpc"/>
<xsd:enumeration value="x86"/>
<xsd:enumeration value="x86_32"/>
<xsd:enumeration value="x86_64"/>
<xsd:enumeration value="parisc"/>
<xsd:enumeration value="mips"/>
<xsd:enumeration value="ia64"/>
<xsd:enumeration value="arm"/>
<xsd:enumeration value="other"/>
</xsd:restriction>
</xsd:simpleType>
<!--============================================================-->
<xsd:simpleType name="OperatingSystemTypeEnumeration">
<xsd:restriction base="xsd:string">
<xsd:enumeration value="Unknown"/>
<xsd:enumeration value="MACOS"/>
<xsd:enumeration value="ATTUNIX"/>
<xsd:enumeration value="DGUX"/>
<xsd:enumeration value="DECNT"/>
<xsd:enumeration value="Tru64_UNIX"/>
<xsd:enumeration value="OpenVMS"/>
<xsd:enumeration value="HPUX"/>
<xsd:enumeration value="AIX"/>
<xsd:enumeration value="MVS"/>
<xsd:enumeration value="OS400"/>
<xsd:enumeration value="OS_2"/>
<xsd:enumeration value="JavaVM"/>
<xsd:enumeration value="MSDOS"/>
<xsd:enumeration value="WIN3x"/>
<xsd:enumeration value="WIN95"/>
<xsd:enumeration value="WIN98"/>
<xsd:enumeration value="WINNT"/>
<xsd:enumeration value="WINCE"/>
<xsd:enumeration value="NCR3000"/>
<xsd:enumeration value="NetWare"/>
<xsd:enumeration value="OSF"/>
```

```
<xsd:enumeration value="DC_OS"/>
<xsd:enumeration value="Reliant_UNIX"/>
<xsd:enumeration value="SCO_UnixWare"/>
<xsd:enumeration value="SCO_OpenServer"/>
<xsd:enumeration value="Sequent"/>
<xsd:enumeration value="IRIX"/>
<xsd:enumeration value="Solaris"/>
<xsd:enumeration value="SunOS"/>
<xsd:enumeration value="U6000"/>
<xsd:enumeration value="ASERIES"/>
<xsd:enumeration value="TandemNSK"/>
<xsd:enumeration value="TandemNT"/>
<xsd:enumeration value="BS2000"/>
<xsd:enumeration value="LINUX"/>
<xsd:enumeration value="Lynx"/>
<xsd:enumeration value="XENIX"/>
<xsd:enumeration value="VM"/>
<xsd:enumeration value="Interactive_UNIX"/>
<xsd:enumeration value="BSDUNIX"/>
<xsd:enumeration value="FreeBSD"/>
<xsd:enumeration value="NetBSD"/>
<xsd:enumeration value="GNU_Hurd"/>
<xsd:enumeration value="OS9"/>
<xsd:enumeration value="MACH_Kernel"/>
<xsd:enumeration value="Inferno"/>
<xsd:enumeration value="QNX"/>
<xsd:enumeration value="EPOC"/>
<xsd:enumeration value="IxWorks"/>
<xsd:enumeration value="VxWorks"/>
<xsd:enumeration value="MiNT"/>
<xsd:enumeration value="BeOS"/>
<xsd:enumeration value="HP_MPE"/>
<xsd:enumeration value="NextStep"/>
<xsd:enumeration value="PalmPilot"/>
<xsd:enumeration value="Rhapsody"/>
<xsd:enumeration value="Windows_2000"/>
<xsd:enumeration value="Dedicated"/>
<xsd:enumeration value="OS_390"/>
<xsd:enumeration value="VSE"/>
<xsd:enumeration value="TPF"/>
<xsd:enumeration value="Windows_R_Me"/>
<xsd:enumeration value="Caldera_Open_UNIX"/>
<xsd:enumeration value="OpenBSD"/>
<xsd:enumeration value="Not_Applicable"/>
<xsd:enumeration value="Windows_XP"/>
<xsd:enumeration value="z_OS"/>
<xsd:enumeration value="other"/>
</xsd:restriction>
</xsd:simpleType>
<!--============================================================-->
<xsd:simpleType name="FileSystemTypeEnumeration">
<xsd:restriction base="xsd:string">
<xsd:enumeration value="swap"/>
<xsd:enumeration value="temporary"/>
<xsd:enumeration value="spool"/>
<xsd:enumeration value="normal"/>
</xsd:restriction>
</xsd:simpleType>
<!--============================================================-->
<xsd:simpleType name="CreationFlagEnumeration">
<xsd:restriction base="xsd:string">
```

```
<xsd:enumeration value="overwrite"/>
<xsd:enumeration value="append"/>
<xsd:enumeration value="dontOverwrite"/>
</xsd:restriction>
</xsd:simpleType>
<!--=========================================================-->
<xsd:simpleType name="Description_Type">
<xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<!--=========================================================-->
<!-- COMPLEX TYPES: Definitions for the RangeValueType -->
<xsd:complexType name="Boundary_Type">
<xsd:simpleContent>
<xsd:extension base="xsd:double">
<xsd:attribute name="exclusiveBound" type="xsd:boolean"
use="optional"/>
<xsd:anyAttribute namespace="##other"
processContents="lax"/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="Exact_Type">
<xsd:simpleContent>
<xsd:extension base="xsd:double">
<xsd:attribute name="epsilon" type="xsd:double"
use="optional"/>
<xsd:anyAttribute namespace="##other"
processContents="lax"/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="Range_Type">
<xsd:sequence>
<xsd:element name="LowerBound" type="Boundary_Type"/>
<xsd:element name="UpperBound" type="Boundary_Type"/>
</xsd:sequence>
<xsd:anyAttribute namespace="##other" processContents="lax"/>
</xsd:complexType>
<xsd:complexType name="RangeValue_Type">
<xsd:sequence>
<xsd:element name="UpperBoundedRange" type="Boundary_Type"
minOccurs="0"/>
<xsd:element name="LowerBoundedRange" type="Boundary_Type"
minOccurs="0"/>
<xsd:element name="Exact" type="Exact_Type" minOccurs="0"
maxOccurs="unbounded"/>
<xsd:element name="Range" type="Range_Type" minOccurs="0"
maxOccurs="unbounded"/>
</xsd:sequence>
<xsd:anyAttribute namespace="##other" processContents="lax"/>
</xsd:complexType>
<!--=========================================================-->
<xsd:complexType name="JobDefinition_Type">
<xsd:sequence>
<xsd:element ref="jsdl:JobDescription"/>
<xsd:any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
</xsd:sequence>
<xsd:attribute name="id" type="xsd:ID" use="optional"/>
<xsd:anyAttribute namespace="##other" processContents="lax"/>
</xsd:complexType>
<!--=========================================================-->
```

```
<xsd:complexType name="JobDescription_Type">
<xsd:sequence>
<xsd:element ref="jsdl:JobIdentification" minOccurs="0"/>
<xsd:element ref="jsdl:Application" minOccurs="0"/>
<xsd:element ref="jsdl:Resources" minOccurs="0"/>
<xsd:element ref="jsdl:DataStaging" minOccurs="0"
maxOccurs="unbounded"/>
<xsd:any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
</xsd:sequence>
<xsd:anyAttribute namespace="##other" processContents="lax"/>
</xsd:complexType>
<!--============================================================-->
<xsd:complexType name="JobIdentification_Type">
<xsd:sequence>
<xsd:element ref="jsdl:JobName" minOccurs="0"/>
<xsd:element ref="jsdl:Description" minOccurs="0"/>
<xsd:element ref="jsdl:JobAnnotation" minOccurs="0"
maxOccurs="unbounded"/>
<xsd:element ref="jsdl:JobProject" minOccurs="0"
maxOccurs="unbounded"/>
<xsd:any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
</xsd:sequence>
<xsd:anyAttribute namespace="##other" processContents="lax"/>
</xsd:complexType>
<!--============================================================-->
<xsd:complexType name="Application_Type">
<xsd:sequence>
<xsd:element ref="jsdl:ApplicationName" minOccurs="0"/>
<xsd:element ref="jsdl:ApplicationVersion" minOccurs="0"/>
<xsd:element ref="jsdl:Description" minOccurs="0"/>
<xsd:any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
</xsd:sequence>
<xsd:anyAttribute namespace="##other" processContents="lax"/>
</xsd:complexType>
<!--============================================================-->
<!-- COMPLEX TYPES : Resource related types -->
<xsd:complexType name="Resources_Type">
<xsd:sequence>
<xsd:element ref="jsdl:CandidateHosts" minOccurs="0"/>
<xsd:element ref="jsdl:FileSystem" minOccurs="0"
maxOccurs="unbounded"/>
<xsd:element ref="jsdl:ExclusiveExecution" minOccurs="0"/>
<xsd:element ref="jsdl:OperatingSystem" minOccurs="0"/>
<xsd:element ref="jsdl:CPUArchitecture" minOccurs="0"/>
<xsd:element ref="jsdl:IndividualCPUSpeed" minOccurs="0"/>
<xsd:element ref="jsdl:IndividualCPUTime" minOccurs="0"/>
<xsd:element ref="jsdl:IndividualCPUCount" minOccurs="0"/>
<xsd:element ref="jsdl:IndividualNetworkBandwidth"
minOccurs="0"/>
<xsd:element ref="jsdl:IndividualPhysicalMemory"
minOccurs="0"/>
<xsd:element ref="jsdl:IndividualVirtualMemory" minOccurs="0"/>
<xsd:element ref="jsdl:IndividualDiskSpace" minOccurs="0"/>
<xsd:element ref="jsdl:TotalCPUTime" minOccurs="0"/>
<xsd:element ref="jsdl:TotalCPUCount" minOccurs="0"/>
<xsd:element ref="jsdl:TotalPhysicalMemory" minOccurs="0"/>
<xsd:element ref="jsdl:TotalVirtualMemory" minOccurs="0"/>
<xsd:element ref="jsdl:TotalDiskSpace" minOccurs="0"/>
```

```
<xsd:element ref="jsdl:TotalResourceCount" minOccurs="0"/>
<xsd:any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
</xsd:sequence>
<xsd:anyAttribute namespace="##other" processContents="lax"/>
</xsd:complexType>
<!--============================================================-->
<xsd:complexType name="CandidateHosts_Type">
<xsd:sequence>
<xsd:element ref="jsdl:HostName" maxOccurs="unbounded"/>
</xsd:sequence>
</xsd:complexType>
<!--============================================================-->
<xsd:complexType name="CPUArchitecture_Type">
<xsd:sequence>
<xsd:element ref="jsdl:CPUArchitectureName"/>
<xsd:any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
</xsd:sequence>
<xsd:anyAttribute namespace="##other" processContents="lax"/>
</xsd:complexType>
<!--============================================================-->
<xsd:complexType name="FileSystem_Type">
<xsd:sequence>
<xsd:element ref="jsdl:FileSystemType" minOccurs="0"/>
<xsd:element ref="jsdl:Description" minOccurs="0"/>
<xsd:element ref="jsdl:MountPoint" minOccurs="0"/>
<xsd:element ref="jsdl:DiskSpace" minOccurs="0"/>
<xsd:any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
</xsd:sequence>
<xsd:attribute name="name" type="xsd:NCName" use="required"/>
<xsd:anyAttribute namespace="##other" processContents="lax"/>
</xsd:complexType>
<!--============================================================-->
<xsd:complexType name="OperatingSystem_Type">
<xsd:sequence>
<xsd:element ref="jsdl:OperatingSystemType" minOccurs="0"/>
<xsd:element ref="jsdl:OperatingSystemVersion" minOccurs="0"/>
<xsd:element ref="jsdl:Description" minOccurs="0"/>
<xsd:any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
</xsd:sequence>
<xsd:anyAttribute namespace="##other" processContents="lax"/>
</xsd:complexType>
<xsd:complexType name="OperatingSystemType_Type">
<xsd:sequence>
<xsd:element ref="jsdl:OperatingSystemName"/>
<xsd:any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
</xsd:sequence>
<xsd:anyAttribute namespace="##other" processContents="lax"/>
</xsd:complexType>
<!--============================================================-->
<!-- Complex types: Data staging -->
<xsd:complexType name="DataStaging_Type">
<xsd:sequence>
<xsd:element ref="jsdl:FileName"/>
<xsd:element ref="jsdl:FilesystemName" minOccurs="0"/>
<xsd:element ref="jsdl:CreationFlag"/>
<xsd:element ref="jsdl:DeleteOnTermination" minOccurs="0"/>
```

```
<xsd:element ref="jsdl:Source" minOccurs="0"/>
<xsd:element ref="jsdl:Target" minOccurs="0"/>
<xsd:any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
</xsd:sequence>
<xsd:attribute name="name" type="xsd:NCName" use="optional"/>
<xsd:anyAttribute namespace="##other" processContents="lax"/>
</xsd:complexType>
<!--============================================================-->
<xsd:complexType name="SourceTarget_Type">
<xsd:sequence>
<xsd:element ref="URI" minOccurs="0"/>
<xsd:any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
</xsd:sequence>
<xsd:anyAttribute namespace="##other" processContents="lax"/>
</xsd:complexType>
<!--============================================================-->
<xsd:element name="JobDefinition" type="jsdl:JobDefinition_Type"/>
<xsd:element name="JobDescription" type="jsdl:JobDescription_Type"/>
<xsd:element name="JobIdentification"
type="jsdl:JobIdentification_Type"/>
<xsd:element name="JobName" type="xsd:string"/>
<xsd:element name="Description" type="jsdl:Description_Type"/>
<xsd:element name="JobAnnotation" type="xsd:string"/>
<xsd:element name="JobProject" type="xsd:string"/>
<xsd:element name="Application" type="jsdl:Application_Type"/>
<xsd:element name="ApplicationName" type="xsd:string"/>
<xsd:element name="ApplicationVersion" type="xsd:string"/>
<xsd:element name="Resources" type="jsdl:Resources_Type"/>
<xsd:element name="CandidateHosts" type="CandidateHosts_Type"/>
<xsd:element name="HostName" type="xsd:string"/>
<xsd:element name="OperatingSystem"
type="jsdl:OperatingSystem_Type"/>
<xsd:element name="OperatingSystemType"
type="jsdl:OperatingSystemType_Type"/>
<xsd:element name="OperatingSystemVersion" type="xsd:string"/>
<xsd:element name="OperatingSystemName"
type="jsdl:OperatingSystemTypeEnumeration"/>
<xsd:element name="ExclusiveExecution" type="xsd:boolean"/>
<xsd:element name="CPUArchitecture"
type="jsdl:CPUArchitecture_Type"/>
<xsd:element name="CPUArchitectureName"
type="jsdl:ProcessorArchitectureEnumeration"/>
<xsd:element name="FileSystem" type="jsdl:FileSystem_Type"/>
<xsd:element name="FileSystemType"
type="jsdl:FileSystemTypeEnumeration"/>
<xsd:element name="MountPoint" type="xsd:string"/>
<xsd:element name="DiskSpace" type="jsdl:RangeValue_Type"/>
<xsd:element name="IndividualCPUTime" type="jsdl:RangeValue_Type"/>
<xsd:element name="IndividualCPUSpeed" type="jsdl:RangeValue_Type"/>
<xsd:element name="IndividualCPUCount" type="jsdl:RangeValue_Type"/>
<xsd:element name="IndividualPhysicalMemory"
type="jsdl:RangeValue_Type"/>
<xsd:element name="IndividualVirtualMemory"
type="jsdl:RangeValue_Type"/>
<xsd:element name="IndividualNetworkBandwidth"
type="jsdl:RangeValue_Type"/>
<xsd:element name="IndividualDiskSpace" type="jsdl:RangeValue_Type"/>
<xsd:element name="TotalCPUTime" type="jsdl:RangeValue_Type"/>
<xsd:element name="TotalCPUCount" type="jsdl:RangeValue_Type"/>
```

```
<xsd:element name="TotalPhysicalMemory" type="jsdl:RangeValue_Type"/>
<xsd:element name="TotalVirtualMemory" type="jsdl:RangeValue_Type"/>
<xsd:element name="TotalDiskSpace" type="jsdl:RangeValue_Type"/>
<xsd:element name="TotalResourceCount" type="jsdl:RangeValue_Type"/>
<xsd:element name="DataStaging" type="jsdl:DataStaging_Type"/>
<xsd:element name="FileName" type="xsd:string"/>
<xsd:element name="FilesystemName" type="xsd:NCName"/>
<xsd:element name="CreationFlag"
type="jsdl:CreationFlagEnumeration"/>
<xsd:element name="DeleteOnTermination" type="xsd:boolean"/>
<xsd:element name="Source" type="jsdl:SourceTarget_Type"/>
<xsd:element name="Target" type="jsdl:SourceTarget_Type"/>
<xsd:element name="URI" type="xsd:anyURI"/>
</xsd:schema>
```

Приложение Б
(обязательное)

Нормативная схема JSDL для приложений POSIX

Настоящий раздел содержит полное определение XML-схемы для JSDL приложений POSIX, версия 1.0.

Для удобства информативная копия этой схемы доступна онлайн по адресу gost.e-ecolog.ru jsdl-posix. Если, однако, будут встречаться отличия между определениями, приведенными в этом разделе, и определениями или их частями, приведенными в других разделах или в онлайн-схеме, нормативными считаются определения, приведенные в настоящем разделе.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
-->
<xsd:schema xmlns:xsd="gost.e-ecolog.ru                    "
xmlns="gost.e-ecolog.ru                    "
xmlns:jsdl-posix="gost.e-ecolog.ru
targetNamespace="gost.e-ecolog.ru
elementFormDefault="qualified">
<!--============================================================-->
<xsd:complexType name="Environment_Type">
<xsd:simpleContent>
<xsd:extension base="xsd:string">
<xsd:attribute name="name" type="xsd:NCName"
use="required"/>
<xsd:attribute name="filesystemName" type="xsd:NCName"
use="optional"/>
<xsd:anyAttribute namespace="##other"
processContents="lax"/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<!--============================================================-->
<xsd:complexType name="Argument_Type">
<xsd:simpleContent>
<xsd:extension base="xsd:normalizedString">
<xsd:attribute name="filesystemName" type="xsd:NCName"
use="optional"/>
<xsd:anyAttribute namespace="##other"
processContents="lax"/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<!--============================================================-->
<xsd:complexType name="FileName_Type">
<xsd:simpleContent>
<xsd:extension base="xsd:string">
<xsd:attribute name="filesystemName" type="xsd:NCName"
use="optional"/>
<xsd:anyAttribute namespace="##other"
processContents="lax"/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<!--============================================================-->
<xsd:complexType name="DirectoryName_Type">
<xsd:simpleContent>
<xsd:extension base="xsd:string">
<xsd:attribute name="filesystemName" type="xsd:NCName"
use="optional"/>
```

```
<xsd:anyAttribute namespace="##other"
processContents="lax"/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<!--=========================================================-->
<xsd:complexType name="Limits_Type">
<xsd:simpleContent>
<xsd:extension base="xsd:nonNegativeInteger">
<xsd:anyAttribute namespace="##other"
processContents="lax"/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<!--=========================================================-->
<xsd:complexType name="UserName_Type">
<xsd:simpleContent>
<xsd:extension base="xsd:string">
<xsd:anyAttribute namespace="##other"
processContents="lax"/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<!--=========================================================-->
<xsd:complexType name="GroupName_Type">
<xsd:simpleContent>
<xsd:extension base="xsd:string">
<xsd:anyAttribute namespace="##other"
processContents="lax"/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<!--=========================================================-->
<xsd:complexType name="POSIXApplication_Type">
<xsd:sequence>
<xsd:element ref="jsdl-posix:Executable" minOccurs="0"/>
<xsd:element ref="jsdl-posix:Argument" minOccurs="0"
maxOccurs="unbounded"/>
<xsd:element ref="jsdl-posix:Input" minOccurs="0"/>
<xsd:element ref="jsdl-posix:Output" minOccurs="0"/>
<xsd:element ref="jsdl-posix:Error" minOccurs="0"/>
<xsd:element ref="jsdl-posix:WorkingDirectory"
minOccurs="0"/>
<xsd:element ref="jsdl-posix:Environment" minOccurs="0"
maxOccurs="unbounded"/>
<xsd:element ref="jsdl-posix:WallTimeLimit" minOccurs="0"/>
<xsd:element ref="jsdl-posix:FileSizeLimit" minOccurs="0"/>
<xsd:element ref="jsdl-posix:CoreDumpLimit" minOccurs="0"/>
<xsd:element ref="jsdl-posix:DataSegmentLimit"
minOccurs="0"/>
<xsd:element ref="jsdl-posix:LockedMemoryLimit"
minOccurs="0"/>
<xsd:element ref="jsdl-posix:MemoryLimit" minOccurs="0"/>
<xsd:element ref="jsdl-posix:OpenDescriptorsLimit"
minOccurs="0"/>
<xsd:element ref="jsdl-posix:PipeSizeLimit" minOccurs="0"/>
<xsd:element ref="jsdl-posix:StackSizeLimit" minOccurs="0"/>
<xsd:element ref="jsdl-posix:CPUTimeLimit" minOccurs="0"/>
<xsd:element ref="jsdl-posix:ProcessCountLimit"
minOccurs="0"/>
<xsd:element ref="jsdl-posix:VirtualMemoryLimit"
```

```
minOccurs="0"/>
<xsd:element ref="jsdl-posix:ThreadCountLimit" minOccurs="0"/>
<xsd:element ref="jsdl-posix:UserName" minOccurs="0"/>
<xsd:element ref="jsdl-posix:GroupName" minOccurs="0"/>
</xsd:sequence>
<xsd:attribute name="name" type="xsd:NCName" use="optional"/>
<xsd:anyAttribute namespace="##other" processContents="lax"/>
</xsd:complexType>
<!--============================================================-->
<xsd:element name="POSIXApplication" type="POSIXApplication_Type"/>
<xsd:element name="Executable" type="jsdl-posix:FileName_Type"/>
<xsd:element name="Argument" type="jsdl-posix:Argument_Type"/>
<xsd:element name="Input" type="jsdl-posix:FileName_Type"/>
<xsd:element name="Output" type="jsdl-posix:FileName_Type"/>
<xsd:element name="Error" type="jsdl-posix:FileName_Type"/>
<xsd:element name="WorkingDirectory"
type="jsdl-posix:DirectoryName_Type"/>
<xsd:element name="Environment" type="jsdl-posix:Environment_Type"/>
<xsd:element name="WallTimeLimit" type="jsdl-posix:Limits_Type"/>
<xsd:element name="FileSizeLimit" type="jsdl-posix:Limits_Type"/>
<xsd:element name="CoreDumpLimit" type="jsdl-posix:Limits_Type"/>
<xsd:element name="DataSegmentLimit" type="jsdl-posix:Limits_Type"/>
<xsd:element name="LockedMemoryLimit" type="jsdl-posix:Limits_Type"/>
<xsd:element name="MemoryLimit" type="jsdl-posix:Limits_Type"/>
<xsd:element name="OpenDescriptorsLimit"
type="jsdl-posix:Limits_Type"/>
<xsd:element name="PipeSizeLimit" type="jsdl-posix:Limits_Type"/>
<xsd:element name="StackSizeLimit" type="jsdl-posix:Limits_Type"/>
<xsd:element name="CPUTimeLimit" type="jsdl-posix:Limits_Type"/>
<xsd:element name="ProcessCountLimit" type="jsdl-posix:Limits_Type"/>
<xsd:element name="VirtualMemoryLimit"
type="jsdl-posix:Limits_Type"/>
<xsd:element name="ThreadCountLimit" type="jsdl-posix:Limits_Type"/>
<xsd:element name="UserName" type="jsdl-posix:UserName_Type"/>
<xsd:element name="GroupName" type="jsdl-posix:GroupName_Type"/>
</xsd:schema>
```

Приложение В
(обязательное)

Нормативная схема JSDL, содержащая сводку исправлений

Настоящее приложение содержит полное нормативное определение XML-схемы исправлений JSDL версии 1.0:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
-->
<xsd:schema
targetNamespace="http://schemas.ggf.org/jsdl/2007/03/jsdl-errata"
xmlns:jsdl-errata="gost.e-ecolog.ru
xmlns:xsd="gost.e-ecolog.ru                    XMLSchema"
elementFormDefault="qualified">
<!-- JSDL errata process schema corrections -->
<!-- ===================================== -->
<!-- artf5497
Filesystem MountSource is not defined in schema -->
<xsd:element name="MountSource" type="xsd:string"/>
</xsd:schema>
```

Приложение Г
(справочное)

Расширенные информационные примеры JSDL

Следующий пример определяет вызов приложения "gnuplot". Пример использует все основные элементы JSDL версии 1.0: JobIdentification, Application, Resources и DataStaging. Данный пример также использует расширение POSIXApplication. Он показывает возможности использования файлов ввода/вывода.

```
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:JobDefinition xmlns="http://www.example.org/"
xmlns:jsdl="gost.e-ecolog.ru                    /jsdl"
xmlns:jsdl-posix="gost.e-ecolog.ru                                    "
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<jsdl:JobDescription>
<jsdl:JobIdentification>
<jsdl:JobName>My Gnuplot invocation</jsdl:JobName>
<jsdl:Description> Simple application invocation
</jsdl:Description>
</jsdl:JobIdentification>
<jsdl:Application>
<jsdl:ApplicationName>gnuplot</jsdl:ApplicationName>
<jsdl-posix:POSIXApplication>
<jsdl-posix:Executable>
/usr/local/bin/gnuplot
</jsdl-posix:Executable>
<jsdl-posix:Argument>control.txt</jsdl-posix:Argument>
<jsdl-posix:Input>input.dat</jsdl-posix:Input>
<jsdl-posix:Output>output1.png</jsdl-posix:Output>
</jsdl-posix:POSIXApplication>
</jsdl:Application>
<jsdl:Resources>
<jsdl:IndividualPhysicalMemory>
<jsdl:LowerBoundedRange>2097152.0</jsdl:LowerBoundedRange>
</jsdl:IndividualPhysicalMemory>
<jsdl:TotalCPUCount>
<jsdl:Exact>1.0</jsdl:Exact>
</jsdl:TotalCPUCount>
</jsdl:Resources>
<jsdl:DataStaging>
<jsdl:FileName>control.txt</jsdl:FileName>
<jsdl:CreationFlag>overwrite</jsdl:CreationFlag>
<jsdl:DeleteOnTermination>true</jsdl:DeleteOnTermination>
<jsdl:Source>
<jsdl:URI>gost.e-ecolog.ru                                    >
</jsdl:Source>
</jsdl:DataStaging>
<jsdl:DataStaging>
<jsdl:FileName>input.dat</jsdl:FileName>
<jsdl:CreationFlag>overwrite</jsdl:CreationFlag>
<jsdl:DeleteOnTermination>true</jsdl:DeleteOnTermination>
<jsdl:Source>
<jsdl:URI>http://foo.bar.com/~me/input.dat</jsdl:URI>
</jsdl:Source>
</jsdl:DataStaging>
<jsdl:DataStaging>
<jsdl:FileName>output1.png</jsdl:FileName>
<jsdl:CreationFlag>overwrite</jsdl:CreationFlag>
<jsdl:DeleteOnTermination>true</jsdl:DeleteOnTermination>
<jsdl:Target>
```

```
<jsdl:URI>rsync://spoolmachine/userdir</jsdl:URI>
</jsdl:Target>
</jsdl:DataStaging>
</jsdl:JobDescription>
</jsdl:JobDefinition>
```

УДК 681.3.06:006.354 ОКС 35.060

Ключевые слова: Грид-среда, Грид-система, интероперабельность, язык, спецификация

Редактор *Л.В. Коретникова*
Технический редактор *В.Н. Прусакова*
Корректор *Е.М. Поляченко*
Компьютерная верстка *Д.В. Кардановской*

Сдано в набор 21.01.2019. Подписано в печать 30.01.2019. Формат 60 × 84 1/8. Гарнитура Ариал.
Усл. печ. л. 7,44. Уч.-изд. л. 6,60.

Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

ИД «Юриспруденция», 115419, Москва, ул. Орджоникидзе, 11.
gost.e-ecolog.ru gost.e-ecolog.ru

Создано в единичном исполнении ФГУП «СТАНДАРТИНФОРМ»
для комплектования Федерального информационного фонда стандартов,
117418 Москва, Нахимовский пр-т, д. 31, к. 2.
gost.e-ecolog.ru stinfo.ru

gost.e-ecolog.ru