



(51) МПК

G06F 9/44 (2006.01)*G06F 9/50* (2006.01)*G06F 17/22* (2006.01)

**ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ,
ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ**

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ(21), (22) Заявка: **2004132573/09**, **09.11.2004**(24) Дата начала отсчета срока действия патента:
09.11.2004(30) Конвенционный приоритет:
09.12.2003 US 10/731,597(43) Дата публикации заявки: **27.04.2006**(45) Опубликовано: **27.10.2009** Бюл. № 30

(56) Список документов, цитированных в отчете о поиске: **PETER VOGEL. New XML Features of the Microsoft Office Word 2003 Object Model, PH&V Information Services**, опуб. октябрь 2003, найдена по адресу URL:[http://msdn.microsoft.com/en-us/library/aa537148\(offic.11,printer\).aspx](http://msdn.microsoft.com/en-us/library/aa537148(offic.11,printer).aspx), разделы "Working with XML" - "Validating Documents". **PAUL CORNER. "Microsoft Office Word 2003 Technical Articles.** (см. прод.)

Адрес для переписки:

**129090, Москва, ул. Б.Спасская, 25, стр.3,
ООО "Юридическая фирма Городисский и
Партнеры", пат.пов. Ю.Д.Кузнецову,
рег.№ 595**

(72) Автор(ы):

**ДЖОНС Брайан М. (US),
САНДЕРЛЭНД Марк (US),
САВИЦКИ Марсин (US),
ЛИТТЛ Роберт А. (US)**

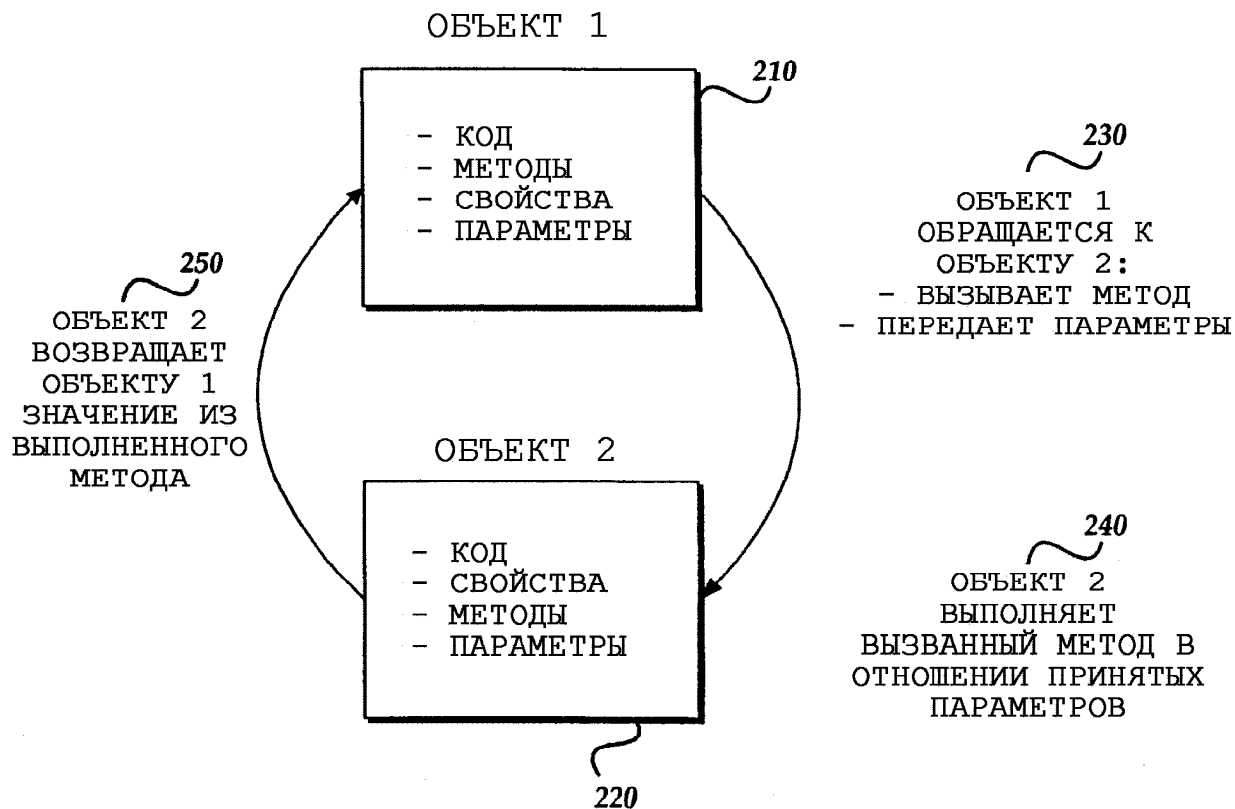
(73) Патентообладатель(и):

МАЙКРОСОФТ КОРПОРЕЙШН (US)**(54) ПРОГРАММИРУЕМАЯ ОБЪЕКТНАЯ МОДЕЛЬ ДЛЯ ПОДДЕРЖКИ БИБЛИОТЕКИ ПРОСТРАНСТВ ИМЕН ИЛИ СХЕМ В ПРОГРАММНОМ ПРИЛОЖЕНИИ**

(57) Реферат:

Изобретение относится к программируемым объектным моделям. Техническим результатом является предоставление пользователю возможности осуществления программным путем доступа и применения библиотеки пространств имен или схем. Указанная библиотека содержит файлы XML-схем и соответствующие XML-ориентированные ресурсы для

связывания файлов схем с одним или более документами и для настройки функциональных возможностей, связанных с файлами схем и XML-ориентированными ресурсами. Упомянутая программируемая объектная модель также предоставляет возможность пользователю/программисту удалять файлы схем и другие XML-ориентированные ресурсы из связи с ранее связанными документами. 3 н. и 12 з.п. ф-лы, 4 ил.



ФИГ. 2

(56) (продолжение):

Microsoft Office Word 2003 XML Object Model Overview", Microsoft Corporation, опуб. Март 2003, найденная по адресу URL:[http://msdn.microsoft.com/en-us/library/aa203683\(offic.11,printer\).aspx](http://msdn.microsoft.com/en-us/library/aa203683(offic.11,printer).aspx), разделы "What you can do with the object model" - "Delet an element". AARON SKINNARD. The XML files. What's new in MSXML 4.0, опуб. декабрь 2001, найденная по адресу URL:[http://msdn.microsoft.com/ru-ru/magazine/cc302178\(en-us,printer\).aspx](http://msdn.microsoft.com/ru-ru/magazine/cc302178(en-us,printer).aspx), разделы "MSXML 4.0 Technology" - "XPath and Type Information". US 2003/0135825 A1 (GERTNER MATTHEW et al), 17.07.2003. НОПЕНКОВ И.П. и др. Телекоммуникационные технологии и сети. - М.: МГТУ им. Н.Э.Баумана, 1998, с.152 - 224.



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY,
PATENTS AND TRADEMARKS

(51) Int. Cl.

G06F 9/44 (2006.01)*G06F 9/50* (2006.01)*G06F 17/22* (2006.01)**(12) ABSTRACT OF INVENTION**(21), (22) Application: **2004132573/09, 09.11.2004**(24) Effective date for property rights:
09.11.2004(30) Priority:
09.12.2003 US 10/731,597(43) Application published: **27.04.2006**(45) Date of publication: **27.10.2009 Bull. 30**

Mail address:

**129090, Moskva, ul. B.Spasskaja, 25, str.3, OOO
"Juridicheskaja firma Gorodisskij i Partnery",
pat.pov. Ju.D.Kuznetsovu, reg.№ 595**

(72) Inventor(s):

**DZhONS Brajan M. (US),
SANDERLEhND Mark (US),
SAVITsKI Marsin (US),
LITTL Robert A. (US)**

(73) Proprietor(s):

MAJKROSOFT KORPOREJShN (US)

(54) PROGRAMMABLE OBJECT MODEL FOR SUPPORTING LIBRARY OF NAME OR SCHEME SPACES IN PROGRAMME APPLICATION

(57) Abstract:

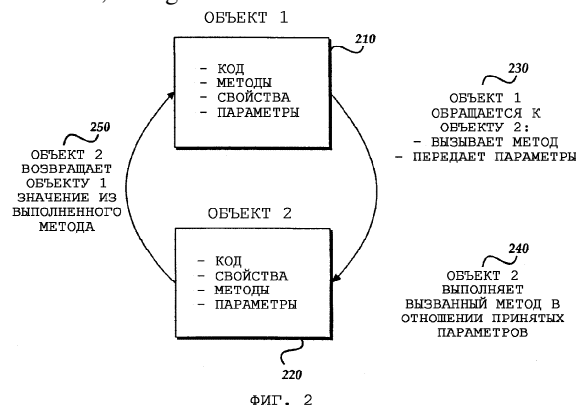
FIELD: information technologies.

SUBSTANCE: library contains XML-scheme files and appropriate XML-oriented resources for connecting the scheme files to one or more documents and for setting the functional capabilities connected to scheme files and XML-oriented resources. The above programmable object model also provides the possibility to user/programmer to delete scheme files and other XML-oriented resources from the connection to the previously connected documents.

EFFECT: providing the user with the possibility of performing in a programmable way the access to

library of name or scheme spaces and using it.

16 cl, 4 dwg



Родственные заявки

Заявка на выдачу патента США на изобретение под номером дела заявителя 60001.0263US01/MS303917.1, озаглавленная «Programmable Object Model for Extensible Markup Language Schema Validation» («Программируемая объектная модель для проверки действительности схемы Расширяемого Языка Разметки (XML)»), и заявка на выдачу патента США на изобретение под номером дела заявителя 60001.0264US01/MS303918.1, озаглавленная «Programmable Object Model for Extensible Markup Language Markup in an Application» («Программируемая объектная модель для XML-разметки в приложении»), включены в настоящее описание посредством ссылки.

Предупреждение об авторских правах

Часть описания этого патентного документа содержит материал, который является объектом охраны авторского права. Владелец авторского права не возражает против факсимильного репродуцирования кем-либо, имеющим отношение к патентному документу или описанию патента, при его появлении в патентном фонде Агентства США по патентам и товарным знакам, но во всех иных случаях абсолютно все любые другие авторские права защищены.

Область техники, к которой относится изобретение

Настоящее изобретение относится к программируемым объектным моделям. Более точно, настоящее изобретение относится к программируемой объектной модели для поддержки библиотеки пространств имен или схем в программном приложении.

Предшествующий уровень техники

Компьютерные программные приложения позволяют пользователям создавать различные документы для помощи им в работе, образовании и досуге. Например, известные приложения электронной обработки текстов позволяют пользователям создавать письма, статьи, книги, заметки и тому подобное. Приложения для обработки электронных таблиц позволяют пользователям сохранять, преобразовывать, печатать и отображать различные буквенно-цифровые данные. Такие приложения имеют множество широко известных возможностей, включая всевозможное редактирование, форматирование, распечатывание, вычисление и интерактивное (онлайновое) и автономное редактирование.

Большинство компьютерных программных приложений не содержат всех необходимых средств программирования для обеспечения функциональных возможностей, требуемых или желаемых всеми потенциальными пользователями. Многие программисты часто хотят улучшить существующие возможности приложения в своих собственных программах или адаптировать функциональность приложения и сделать ее более пригодной для обособленного множества пользователей или действий. Например, программист, работающий в финансовой индустрии, может пожелать переделать текстовый обработчик для круга пользователей, состоящего из финансовых аналитиков, редактирующих финансовые отчеты. В последние годы Расширяемый Язык Разметки (XML) широко использовался как взаимозаменяемый формат данных для многих пользователей. Часто пользователи функциональных возможностей XML присоединяют или ассоциируют один или более файлов XML-схем или XML-ориентированных решений с документом, находящимся в состоянии редактирования или создания пользователем. Однако пользователи и/или программисты ограничены в своих возможностях применять функциональность файлов XML-схем или XML-ориентированных решений для данного документа, потому что пользователь и/или программист не имеет прямого и

легкого доступа к библиотеке пространств имен или схем, содержащей файлы XML-схем или другие XML-ориентированные решения.

Следовательно, в данной области техники существует необходимость в программируемой объектной модели для предоставления пользователю/программисту возможности получать доступ к библиотеке пространств имен и схем, состоящей из XML ресурсов, для конфигурирования на индивидуальной основе или иного манипулирования такими ресурсами, чтобы расширить использование пользователем/программистом функциональных возможностей XML совместно с документом программного приложения. Именно по этим и другим соображениям было сделано настоящее изобретение.

Сущность изобретения

Настоящее изобретение обеспечивает способы и системы для предоставления пользователю возможности осуществления программным путем доступа и применения библиотеки пространств имен или схем, содержащей файлы XML-схем и соответствующие XML-ориентированные ресурсы, для связывания этих XML-ресурсов с одним или большим количеством документов и для настройки функциональных возможностей, связанных с такими XML-ориентированными ресурсами. Как только пользователь или программист получает доступ к библиотеке пространств имен и схем, такой пользователь имеет возможность программным путем связывать файлы XML-схем с XML-данными в связанном с ними документе и, наоборот, пользователь может обнаруживать и удалять связи файлов XML-схем с XML-данными, содержащимися в упомянутом документе. Пользователь может также программным путем связывать файлы преобразований с XML-данными, содержащимися в документе, и обнаруживать и удалять существующие файлы преобразований, связанные с XML-данными, содержащимися в этом документе. Пользователь может также связывать другие файлы и исполняемое программное обеспечение, связанное с XML-ориентированным и другими решениями документов, с XML-данными, содержащимися в документе. Дополнительно пользователь может обнаруживать и удалять связь XML-ориентированных решений и других типов исполняемого программного обеспечения из связи с XML-данными, содержащимися в документе.

Эти и другие признаки, преимущества и аспекты настоящего изобретения могут быть более ясно поняты и оценены из обзора последующего подробного описания раскрытых воплощений и посредством обращения к прилагаемым чертежам и формуле изобретения.

Перечень фигур чертежей

Фиг.1 - упрощенная блок-схема вычислительной системы и связанных с ней периферийного оборудования и сетевых устройств, которые обеспечивают иллюстративную рабочую среду для настоящего изобретения.

Фиг.2 - упрощенная блок-схема, иллюстрирующая взаимодействие между программными объектами в соответствии с объектно-ориентированной программной моделью.

Фиг.3 - блок-схема, иллюстрирующая взаимодействие между документом, присоединенным файлом схемы, и моделью функциональных возможностей проверки действительности схемы.

Фиг.4 - блок-схема, иллюстрирующая взаимодействие между документом, библиотекой пространств имен или схем и сторонним программным приложением.

Подробное описание предпочтительного воплощения

Воплощения настоящего изобретения направлены на способы и системы для предоставления пользователю возможности программным путем вызывать библиотеку пространств имен/схем, состоящую из файлов XML-схем и XML-ориентированных решений и ресурсов, для управления связыванием таких файлов, решений и ресурсов с одним или большим количеством документов. Такие воплощения могут комбинироваться, могут использоваться другие воплощения и могут производиться структурные изменения, не выходя за рамки сущности и объема настоящего изобретения. Последующее подробное описание, таким образом, не следует интерпретировать в смысле ограничения, и объем настоящего изобретения определяется прилагаемой формулой изобретения и ее эквивалентами.

Далее, ссылаясь на чертежи, на которых идентичные номера соответствуют одинаковым элементам на нескольких чертежах, будут описаны аспекты настоящего изобретения и иллюстративная операционная среда. Фиг.1 и последующее подробное обсуждение предназначены для предоставления краткого, общего описания подходящей вычислительной среды, в которой изобретение может быть реализовано. Хотя настоящее изобретение будет описываться в общем контексте программных модулей, которые выполняются в совокупности с прикладной программой, исполняющейся под управлением операционной системы на персональном компьютере, специалистам в этой области техники будет понятно, что изобретение может также быть реализовано в сочетании с другими программными модулями.

Обычно программные модули включают в себя процедуры, программы, компоненты, структуры данных и другие типы структур, которые выполняют конкретные задачи или реализуют определенные абстрактные типы данных. Кроме того, специалистам в данной области техники должно быть понятно, что изобретение может быть реализовано на практике с другими конфигурациями компьютерных систем, включая карманные устройства, мультипроцессорные системы, микропроцессорную или программируемую бытовую электронику, миникомпьютеры, универсальные вычислительные машины (мейнфреймы) и тому подобное. Данное изобретение может также быть реализовано в распределенных вычислительных средах, где задачи выполняются удаленными устройствами обработки данных, которые соединены через сеть связи. В распределенной вычислительной среде программный модуль может располагаться как на локальных, так и на удаленных запоминающих устройствах.

Далее со ссылкой на фиг.1 будет описана иллюстративная компьютерная архитектура для персонального компьютера 2, предназначенного для практической реализации разнообразных воплощений этого изобретения. Компьютерная архитектура, показанная на фиг.1, изображает обычный персональный компьютер, включающий в себя центральное процессорное устройство 4 (ЦПУ), системную память 6, включающую в себя оперативное запоминающее устройство 8 (ОЗУ) и постоянное запоминающее устройство 10 (ПЗУ), и системную шину 12, которая соединяет упомянутую память с ЦПУ 4. Базовая система ввода/вывода, содержащая основные процедуры, которые помогают передавать информацию между элементами внутри компьютера, например, во время запуска, хранится в ПЗУ 10. Персональный компьютер 2 дополнительно включает в себя запоминающее устройство 14 большой емкости для хранения операционной системы 16, прикладных программ, таких как прикладная программа 305, и данных.

Запоминающее устройство 14 большой емкости подключено к ЦПУ 4 через контроллер запоминающего устройства большой емкости (не показан),

подключенный к шине 12. Запоминающее устройство 14 большой емкости и связанный с ним машиночитаемый носитель обеспечивают энергонезависимое хранилище для персонального компьютера 2. Хотя описание машиночитаемого носителя, содержащееся в материалах настоящей заявки, относится к запоминающим устройствам большой емкости, таким как жесткий диск или дисковод для компакт-дисков, специалистам в данной области техники должно быть понятно, что машиночитаемый носитель может быть любым имеющимся в наличии носителем, к которому персональный компьютер 2 может осуществить доступ.

В качестве примера, но не ограничения, машиночитаемые носители могут содержать компьютерные носители информации или среды передачи информации. Компьютерные носители информации включают в себя энергонезависимые и энергонезависимые, съемные и несъемные носители, реализованные любым способом или по любой технологии для хранения информации, такой как машиночитаемые команды, структуры данных, программные модули или другие данные. Компьютерные носители информации включают в себя, но не в ограничительном смысле, ОЗУ, ПЗУ, электрически программируемое ПЗУ (EPROM), электрически стираемое программируемое ПЗУ (EEPROM), флэш-память или другую память твердотельной технологии, ПЗУ на компакт-диске (CD-ROM), цифровой многофункциональный диск (DVD) или другой оптический накопитель, магнитные дискеты, магнитную ленту, накопитель на магнитных дисках или другие магнитные запоминающие устройства либо любой другой носитель, который может использоваться для хранения желаемой информации и к которому компьютер может осуществить доступ.

В соответствии с разнообразными воплощениями изобретения персональный компьютер 2 может работать в сетевой среде, используя логические соединения с удаленными компьютерами через сеть 18 протокола управления передачей/межсетевого протокола (TCP/IP), такую как Интернет. Персональный компьютер 2 может подключаться к сети 18 TCP/IP через устройство 20 сетевого интерфейса, подключенное к шине 12. Должно быть принято во внимание, что устройство 20 сетевого интерфейса может также использоваться для подключения к другим типам сетей и удаленных компьютерных систем. Персональный компьютер 2 может также включать в себя контроллер 22 ввода/вывода для приема и обработки входной информации от некоторого количества устройств, включая клавиатуру или мышь (не показаны). Подобным образом контроллер 22 ввода/вывода может обеспечивать вывод на экран дисплея, принтер или другой тип устройства вывода.

Как коротко упомянуто выше, некоторое количество программных модулей и файлов данных могут храниться в запоминающем устройстве 14 большой емкости и ОЗУ 8 из состава персонального компьютера 2, включая операционную систему 16, подходящую для управления работой подключенного в сеть персонального компьютера, такую как операционная система WINDOWS XP, разработанная компанией MICROSOFT CORPORATION, г. Редмонд, штат Вашингтон. Запоминающее устройство 14 большой емкости и ОЗУ 8 могут также хранить одну или большее количество прикладных программ. В частности, запоминающее устройство 14 большой емкости и ОЗУ 8 могут хранить прикладную программу 305 для создания и редактирования электронного документа 310. Например, прикладная программа 305 может заключать в себе прикладную программу обработки текстов, приложение обработки электронных таблиц, приложение управления контактной информацией или подобное. Прикладные программы для создания и редактирования других типов

электронных документов могут также быть использованы с разнообразными воплощениями настоящего изобретения. Файл 330 схемы и библиотека 400 пространств имен/схем, описанные ниже, также показаны.

Иллюстративные воплощения настоящего изобретения реализованы посредством обмена данными между различными программными объектами в объектно-ориентированном программном окружении. В целях последующего описания воплощений настоящего изобретения полезно коротко описать компоненты объектно-ориентированного программного окружения. Фиг.2 - это упрощенная блок-схема, иллюстрирующая взаимодействие между программными объектами согласно модели объектно-ориентированного программирования. В соответствии с объектно-ориентированным программным окружением первый объект 210 может включать в себя программный код, исполняемые методы, свойства и параметры. Подобным образом, второй объект 220 может также содержать программный код, исполняемые методы, свойства и параметры.

Первый объект 210 может связываться со вторым объектом 220 для получения информации или функциональных возможностей от второго объекта 220 посредством вызова второго объекта 220 через привязанный к сообщению ызов 230. Как хорошо известно специалистам данной области техники, первый объект 210 может общаться со вторым объектом 220 через программный интерфейс приложения (API), который позволяет двум разнородным программным объектам 210 и 220 общаться друг с другом для получения информации и функциональных возможностей друг от друга. Например, если первому объекту 210 требуются функциональные возможности, предоставляемые методом, содержащимся во втором объекте 220, первый объект 210 может послать привязанный к сообщению вызов 230 второму объекту 220, в котором первый объект идентифицирует такой требуемый метод и в котором первый объект посылает любые требуемые параметры второму объекту, необходимые второму объекту для управления идентифицированным методом. Как только второй объект 220 принимает вызов от первого объекта, второй объект выполняет вызываемый метод на основе предоставленных параметров и посылает возвращаемое сообщение 250, содержащее значение, полученное от выполненного метода, обратно первому объекту 210.

Например, в терминах воплощений настоящего изобретения, и как будет описано далее, первый объект 210 может быть сторонним заказным приложением, которое посылает сообщение второму объекту, такому как объект проверки действительности схемы Расширяемого Языка Разметки (XML-схемы), в соответствии с чем первый объект идентифицирует метод, требуя проверки действительности заданного XML-элемента в документе, где заданный XML-элемент - это параметр, передаваемый первым объектом совместно с идентифицированным методом. После приема такого вызова от первого объекта, согласно этому примеру, объект проверки действительности схемы исполняет идентифицированный метод в отношении заданного XML-элемента и возвращает сообщение первому объекту в форме результата или значения, ассоциированного с прошедшим проверку действительности XML-элементом. Действие объектно-ориентированного программного окружения, такого как коротко описано выше, хорошо известно специалистам в данной области техники.

Как описано ниже, воплощения настоящего изобретения реализуются через взаимодействие программных объектов при использовании, конфигурировании на индивидуальной основе и объявлении компонентов Расширяемого Языка

Разметки (XML). Фиг.3 - это блок-схема, иллюстрирующая взаимодействие между документом, присоединенным файлом XML-схемы и модулем функциональных возможностей проверки действительности схемы. Как хорошо известно специалистам в данной области техники, Расширяемый Язык Разметки (XML) предоставляет способ

описания текста и данных в документе путем предоставления пользователю возможности создавать имена тегов (неотображаемых элементов разметки), которые применены к тексту или данным в документе, который в свою очередь определяет текст или данные, к которым применяются ассоциированные с ними теги. Например, согласно фиг.3, документ 310, созданный посредством приложения 305, содержит текст, который был размечен XML-тегами 315, 320 и 325. Например, текст «Приветствую» аннотирован XML-тегом <title> (<заглавие>). Текст «Меня зовут Сара» аннотирован тегом <body> (<тело>). В соответствии с XML создатель тегов <title> и <body> волен создавать его собственные теги для описания тех тегов, к которым упомянутые теги будут применяться. Затем при условии, что любому приложению-потребителю или вычислительной машине-потребителю предоставлены инструкции в отношении определения тегов, применяемых к тексту, такое приложение или вычислительная машина способны использовать упомянутые данные в соответствии с тегами. Например, если приложение-потребитель запрограммировано извлекать текст, определенный как заглавия статей или публикаций, обрабатываемых таким приложением, то это приложение может анализировать документ 310 и извлекать текст «Приветствую», как показано на фиг.3, потому что такой текст аннотирован тегом <title>. Создатель именованных конкретных XML-тегов для документа 310, изображенного на фиг.3, обеспечивает полезное описание для текста или данных, содержащихся в документе 310, которое может использоваться представителями третьей стороны при условии, что этим представителям третьей стороны предоставлены определения, связанные с тегами, примененными к этому тексту или данным.

В соответствии с воплощениями настоящего изобретения текст и XML-разметку, введенные в документ 310, можно сохранять согласно множеству различных файловых форматов и в соответствии с собственным языком программирования приложения 305, при помощи которого документ 310 создан. Например, текст и XML-разметку можно сохранять согласно приложению электронной обработки текстов, приложению обработки электронных таблиц и им подобным. В качестве альтернативы, текст и XML-разметку, введенные в документ 310, можно сохранять в формате XML, посредством чего текст или данные, любая применяемая XML-разметка и любое форматирование, такое как шрифт, стиль, структура абзацев и т.д., могут сохраняться в XML-представлении. Следовательно, приложение-потребитель или стороннее приложение, выполненное с возможностью понимания данных, сохраненных как XML, могут открыть и обработать текст или данные, сохраненные в XML-представлении. Для детального всестороннего исследования сохранения текста и XML-разметки, а также ассоциированного форматирования и других атрибутов документа 310 в формате XML обратитесь к заявке на патент США, озаглавленной «Word Processing Document Stored in a Single XML File that may be Manipulated by Applications that Understanding XML» («Документ обработки текстов, сохраняемый в едином XML файле, которым могут манипулировать совместимые с XML приложения», серийный номер 10/187060, поданной 28 июня 2002 года, которая целиком включена в материалы настоящей заявки в качестве ссылки.

Для того чтобы предоставить оболочку области определения для элементов

XML-разметки (тегов), примененных к тексту или данным, как показано на фиг.3, создаются файлы XML-схем, которые содержат информацию, необходимую для предоставления пользователям и потребителям размеченных и сохраненных данных возможности понимать определения тегирования XML, разработанные создателем документа. Каждый файл схемы, также упоминаемый в данной области техники как файл определения схемы XML (XSD), предпочтительно включает в себя список всех XML-элементов (тегов), которые могут быть применены к документу согласно данному файлу схемы. Например, файл 330 схемы, показанный на фиг.3, может являться файлом схемы, содержащим определения некоторых XML-элементов, которые могут применяться к документу 310, включая атрибуты XML-элементов или ограничения и/или правила, связанные с текстом или данными, которые могут аннотироваться XML-элементами в соответствии с файлом схемы. Например, обращаясь к файлу 330 схемы, показанному на фиг.3, упомянутый файл схемы идентифицируется пространством имен «intro» и этот же файл схемы включает в себя корневой элемент <introCard>.

В соответствии с файлом 330 схемы такой элемент <introCard> служит как корневой элемент для файла схемы и также как родительский элемент для двух дочерних элементов <title> и <body>. Как хорошо известно специалистам в данной области техники, множество родительских элементов может быть определено под единым корневым элементом и множество дочерних элементов может быть определено под каждым родительским элементом. В типичном виде, однако, заданный файл 330 схемы содержит только один корневой элемент. Ссылаясь еще раз на фиг.3, файл 330 схемы также содержит атрибуты 340 и 345, относящиеся к элементам <title> и <body> соответственно. Такие атрибуты 340 и 345 могут обеспечивать дополнительное определение или правила, связанные с применением соответствующих элементов к тексту и данным документа 310. Например, атрибут 345 определяет, что текст, аннотированный элементом <title>, должен быть не более двадцати пяти символов в длину. Следовательно, если текст, превышающий двадцать пять символов в длину, аннотируется элементом или тегом <title>, пробное аннотирование такого текста будет ошибочным в соответствии с определениями, содержащимися в файле 330 схемы.

Применяя такие определения или правила к XML-элементам в качестве атрибутов, создатель схемы может задавать структуру данных, содержащихся в документе, связанном с заданным файлом схемы. Например, если создатель файла 330 схемы для определения XML-разметки, применяемой к документу резюме, желает, чтобы раздел опыта работы в документе резюме содержал не более четырех записей о настоящей или предшествующей работе, то создатель файла 330 схемы может задать атрибут элемента <experience> (<опыт работы>), например, с целью обеспечения того, чтобы не более четырех записей о настоящем и предшествующих местах работы могло вводиться между тегами <experience>, чтобы текст в отношении опыта работы был действительным в соответствии с файлом 330 схемы. Как хорошо известно специалистам в данной области техники, файл 330 схемы может быть присоединен или иным образом ассоциирован с заданным документом 310 для применения допустимой XML-разметки, определенной в присоединенном к документу 310 файле схемы. Согласно этому же воплощению документ 310, размеченный посредством элементов XML-схемы присоединенного или ассоциированного файла 330 схемы, может указывать на этот присоединенный или ассоциированный файл схемы посредством указания на универсальный идентификатор ресурса (URI), связанный с пространством имен, идентифицирующим присоединенный или ассоциированный

файл 330 схемы.

Согласно воплощениям настоящего изобретения документ 310 может иметь множество присоединенных файлов схем. То есть создатель документа 310 может ассоциативно связывать или присоединять более одного файла 330 схемы к документу 310 для того, чтобы предоставить оболочку для аннотации с помощью XML-разметки из более чем одного файла схемы. Например, документ 310 может содержать текст или данные, связанные с финансовыми данными. Создатель документа 310 может пожелать ассоциативно связать файлы 330 XML-схем, содержащие разметку и определения, связанные с множеством финансовых учреждений. Следовательно, создатель документа 310 может ассоциативно связать файл XML-схемы из одного или более финансовых учреждений с документом 310. Подобным образом, заданный файл 330 XML-схемы может быть ассоциирован с конкретной структурой документа, такой как шаблон для размещения финансовых данных в желаемом формате.

В соответствии с воплощениями настоящего изобретения коллекция файлов XML-схем и ассоциированных решений документов может поддерживаться в библиотеке пространства имен или схем, размещенной отдельно от документа 310. Документ 310, в свою очередь, может содержать указатели на универсальные идентификаторы ресурсов (URI) в библиотеке пространства имен или схем, связанной с одним или более файлов схем, присоединенных или иным образом ассоциированных с документом 310. Так как документ 310 требует информации из одного или более ассоциированных файлов схем, этот документ 310 обращается к библиотеке пространств имен или схем, чтобы получить требуемые определения схем. Для детального описания такого использования операций библиотеки пространств имен или схем обратитесь к заявке на патент США, озаглавленной «System and Method for Providing Namespace Related Information» («Система и способ для предоставления связанной с пространством имен информации»), серийный номер 10/184190, поданной 27 июня 2002 года, и к заявке на патент США, озаглавленной «System and Method for Obtaining and Using Namespace Related Information for Opening XML Documents» («Система и способ для получения и использования связанной с пространством имен информации для открытия XML-документов»), серийный номер 10/185940, поданной 27 июня 2002 года, причем обе заявки на патент США полностью включены в настоящее описание посредством ссылки. Для детального описания алгоритма для загрузки программных компонентов, таких как файлы XML-схем и ассоциированные решения, из библиотеки пространств имен или схем обратитесь к заявке на патент США, озаглавленной «Mechanism for Downloading Software Components from a Remote Source for Use by a Local Software Application» («Алгоритм для загрузки программных компонентов из удаленного источника с помощью локального программного приложения»), серийный номер 10/164260, поданной 5 июня 2002 года.

Согласно фиг.3 модуль функциональных возможностей проверки действительности схемы 350 приведен в качестве иллюстрации проверки действительности XML-разметки, примененной к документу 310, по отношению к файлу 330 XML-схемы, присоединенному или иным образом ассоциированному с документом 310, как описано выше. Как описано выше, файл 330 схемы устанавливает допустимые XML-элементы и ассоциированные атрибуты и определяет правила для действительной аннотации документа 310 XML-разметкой из ассоциированного файла 330 схемы. Например, как показано в файле 330 схемы, два дочерних элемента

<title> и <body> определены под корневым или родительским элементом <introCard>. Также показаны атрибуты 340, 345, задающие допустимую длину строки текста, связанного с дочерними элементами <title> и <body>. Как описано выше, если пользователь попытается аннотировать документ 310 XML-разметкой из файла 330 5
схемы, присоединенного или ассоциированного с документом, в противоречие определениям XML- разметки, содержащимся в файле 330 схемы, то будет возникать недопустимое или ошибочное состояние. Например, если пользователь попытается ввести строку заголовка, длина которой превышает двадцать пять символов, то такая 10
текстовая строка будет противоречить атрибуту максимальной символьной длины элемента <title> файла 330 схемы. Для того чтобы удостовериться в действительности XML-разметки, примененной к документу 310, в отношении ассоциированного файла 330 схемы, используется модуль 350 проверки действительности схемы. Как должно быть понято специалистами в данной области 15
техники, модуль 350 проверки действительности схемы - это программный модуль, включающий в себя машиноисполняемые команды, достаточные для сравнения XML-разметки и ассоциированного текста, введенных в документ, с ассоциированным или присоединенным файлом 330 XML-схемы по мере введения XML-разметки и 20
ассоциированного текста в документ 310.

Согласно воплощениям настоящего изобретения модуль 350 проверки действительности схемы сравнивает каждый элемент XML-разметки и ассоциированного текста или данных, отнесенных к документу 310, с присоединенным 25
или ассоциированным файлом 330 схемы, чтобы определить, соответствует ли каждый элемент и ассоциированный текст или данные правилам и определениям, установленным присоединенным файлом 330 схемы. Например, если пользователь попытается ввести символьную строку, длина которой превышает двадцать пять символов, аннотированных элементами 320 <title>, модуль проверки действительности 30
схемы сопоставит такую текстовую строку с атрибутом 340 текстовой строки присоединенного файла 330 схемы и определит, что длина текстовой строки, введенной пользователем, превышает максимальную допустимую длину текстовой строки. Следовательно, сообщение или диалог с информацией об ошибке будут предоставлены пользователю, чтобы известить пользователя о том, что длина 35
вводимой пользователем строки превышает максимально допустимую длину в символах, соответствующую присоединенному файлу 330 схемы. Подобным образом, если пользователь попытается добавить элемент XML-разметки между элементами <title> и <body>, то модуль 350 проверки действительности определит, что 40
элемент XML-разметки, примененный пользователем, не является действительным элементом, допустимым между элементами <title> и <body>, согласно присоединенному файлу 330 схемы. Следовательно, модуль 350 проверки действительности сгенерирует сообщение или диалог с информацией об ошибке, чтобы известить пользователя о недействительности XML-разметки.

45 Программируемая объектная модель для библиотек пространств имен или схем

Как описано выше со ссылкой на фиг.3, для того чтобы обеспечить оболочку области определения, ориентированную на правила для применения разметки 50
Расширяемого Языка Разметки (XML) к документу 310, один или более файлов 330 схем могут быть присоединены или ассоциированы с упомянутым документом для задания для документа 310 определений и правил, управляющих применением элементов XML-разметки, соответствующих заданному файлу 330 схемы. Как описано, множество файлов XML-схем и других решений документов, например,

предварительно структурированные шаблоны, могут быть присоединены или ассоциированы с одним XML документом 310. Более того, как описано выше, множество различных файлов XML-схем, идентифицированных идентификаторами пространства имен, и множество решений документа могут сохраняться в библиотеке

5 пространств имен или схем отдельно от документа 310. В соответствии с воплощениями настоящего изобретения пользователям дается возможность программного вызова библиотеки пространств имен или схем, связанной с одним или более документами 310, для настройки или иного манипулирования пространствами

10 имен файлов схем и ассоциированными определениями, правилами, ресурсами, а также решениями, связанными с различными идентификаторами пространств имен, содержащимися в библиотеке пространств имен или схем.

Фиг.4 - это блок-схема, иллюстрирующая взаимодействие между документом 310, библиотекой 400 пространств имен или схем и сторонним приложением 450. Согласно

15 воплощениям настоящего изобретения пользователи могут программно вызывать библиотеку 400 пространств имен через совокупность объектно-ориентированных привязанных к сообщениям вызовов или программных интерфейсов 470 приложений для изменения содержимого или действия отдельных файлов 410, 430 схем или ресурсов 420, 440, связанных с файлами схем, идентифицированными в библиотеке 400

20 пространства имен. Пользователь может взаимодействовать с библиотекой 400 пространств имен из приложения 305 или из сторонней программы через совокупность объектно-ориентированных привязанных к сообщениям вызовов, и сторонняя программа может быть разработана с использованием разнообразных

25 языков программирования, таких как C, C++, C#, Visual Basic и им подобных.

Имея доступ к библиотеке пространств имен через набор программных интерфейсов 470 приложений, пользователь может программным путем связывать один или более дополнительных файлов XML-схем или пространств имен с

30 XML-данными, и наоборот, пользователь может обнаруживать и удалять существующие связи между одним или более файлов XML-схем и XML-данными или разметкой, примененной к документу 310. Пользователь может также программным путем связывать Преобразование Расширяемого Языка Таблиц Стилей (XSLT) с XML-данными, примененными к документу, и наоборот, пользователь может

35 обнаруживать и удалять существующие XSLT-преобразования из связи с XML-данными, примененными к документу 310. Более того, пользователь может программным путем связывать другие файлы и исполняемые программные приложения с XML-данными, примененными к документу 310, и определять и удалять

40 существующие связи других программных приложений и файлов с XML-данными.

Например, пространство 430 имен, показанное в библиотеке 400 пространств имен, может содержать решение, состоящее из предварительно отформатированной структуры для шаблона документа резюме. Когда такое решение применяется к документу 310, ассоциированные определения схемы и правила, разработанные

45 создателем шаблона документа резюме, будут применяться к XML-разметке и ассоциированному тексту, введенным в документ 310. Если файл схемы, связанный с шаблоном документа резюме, требует, чтобы раздел опыта работы документа резюме содержал по меньшей мере три описания последней или настоящей работы, такое

50 определение схемы будет применяться к документу 310 таким образом, что по меньшей мере три описания работы должны быть введены последующим пользователем в секцию опыта работы для того, чтобы такой XML-документ 310 был признан действительным модулем 350 проверки действительности схемы. В

продолжение этого примера, если такой файл схемы шаблона документа резюме связан с документом 310, и пользователь желает удалить связь этого файла схемы с документом 310, то пользователь может действовать также программным путем из сторонней программы, посылая объектно-ориентированный привязанный к сообщению вызов в библиотеку 450 пространств имен или в приложение 305 посредством поддерживаемого программного интерфейса приложения, для управления удалением связи файла схемы шаблона документа резюме из документа 310.

Нижеследующее является описанием объектов и ассоциированных свойств, включая объектно-ориентированные привязанные к сообщениям вызовы или программные интерфейсы приложений, которые предоставляют пользователю возможность программным путем осуществлять доступ к библиотеке пространств имен, как было описано выше. Для каждого из объектов и связанных с ним свойств ниже приведено описание работы и набора функций упомянутого объекта или ассоциированного свойства.

Объект Application (Приложение)

Далее приведены методы и свойства объекта.

Свойство .XMLNamespaces (Пространства имен XML)

Доступный только на чтение указатель на коллекцию XMLNamespaces пространств имен XML, который представляет библиотеку пространств имен, доступную приложению.

Объект XMLNamespaces collection (коллекция пространств имен XML)

- объект, предоставляющий доступ к объектам XMLNamespace (пространствам имен XML). Он представляет библиотеку пространств имен. Каждый объект XMLNamespace в коллекции представляет одиночное и уникальное пространство имен в библиотеке пространств имен. Ниже приведены методы и свойства такого объекта.

Метод .Add() (добавить)

Метод, создающий и добавляющий к коллекции новый объект XMLNamespace. Он используется для регистрации нового пространства имен в библиотеке пространств имен. Он возвращает новый объект XMLNamespace. Он может принимать следующие параметры.

Path (путь) - указатель на файл схемы для пространства имен. Упомянутый указатель может быть путем файла, представленным как строка (string).

NamespaceURI (универсальный идентификатор ресурса пространства имен) - URI упомянутого пространства имен, которое представляет схему. URI может быть текстовой строкой.

Alias (Псевдоним) - текстовая строка, представляющая альтернативное (более дружественное пользователю) имя для пространства имен, которое может быть задано программистом.

InstallForAllUsers (Установить для всех пользователей) - флаг, показывающий, должно ли новое пространство имен в библиотеке пространств имен быть доступно всем пользователям компьютера либо только текущему пользователю.

Свойство .Application (приложение)

Доступный только на чтение указатель на объект Application, представляющий приложение этой объектной модели.

Свойство .Count (счет)

Доступное только на чтение свойство, возвращающее количество

зарегистрированных пространств имен в библиотеке пространств имен. Это свойство идентично общему количеству объектов XMLNamespace в коллекции пространств имен XML.

Свойство .Creator (создатель)

Доступный только на чтение указатель на создателя этого объекта.

Метод .InstallManifest() (установить манифесты)

Метод для установки манифестов (XML-файлов, описывающих системное окружение исполняемого модуля) решения, которые регистрируют пространства имен в библиотеке пространств имен. Он может принимать следующие параметры.

Path - указатель на файл манифеста для упомянутого манифеста. Этот указатель может быть путем файла, представленным текстовой строкой.

InstallForAllUsers - флаг, показывающий, должны ли быть доступны новые пространства имен, установленные упомянутым манифестом в библиотеке пространств имен, всем пользователям компьютера либо только текущему пользователю.

Метод .Item() (Получить компоненту)

Метод для осуществления доступа к отдельным членам коллекции, использующий числовой индекс или ключевое слово поиска. Такой метод может принимать следующий параметр.

Index (индекс) - номер, представляющий расположение запрашиваемого объекта XMLNamespace в библиотеке пространств имен. Такой индекс может также быть текстовой строкой, представляющей псевдоним или URI запрашиваемого пространства имен.

Свойство .Parent (родитель)

Доступное только на чтение свойство, возвращающее родительский объект коллекции. Это свойство возвращает указатель на приложение, из которого осуществляется доступ к коллекции XMLNamespaces.

Объект XMLNamespace (пространство имен XML) - объект, представляющий отдельную компоненту пространства имен в библиотеке пространств имен (и отдельный элемент в коллекции XMLNamespaces). Ниже приведены методы и свойства такого объекта.

Свойство .Alias (псевдоним)

Свойство для управления псевдонимом, который программист связывает с пространством имен. Оно может поддерживать следующий параметр.

AllUsers (все пользователи) - флаг, показывающий, доступен ли такой псевдоним всем пользователям либо только текущему пользователю.

Свойство .Application (приложение)

Доступный только на чтение указатель на объект Application, представляющий приложение этой объектной модели.

Метод .AttachToDocument() (присоединить к документу)

Метод для присоединения схемы пространства имен, представленного упомянутым объектом, к выбранному документу. Он поддерживает следующий параметр:

Document (документ) - указатель на документ, к которому требуется присоединить упомянутую схему.

Свойство .Creator (создатель)

Доступный только на чтение указатель на создателя упомянутого объекта.

Свойство .DefaultTransform (преобразование по умолчанию)

Свойство, которое указывает на XSLT-преобразование по умолчанию, связанное с

этим пространством имен. Оно может поддерживать следующий параметр.

AllUsers (все пользователи) - флаг, показывающий, должна ли установка преобразования по умолчанию воздействовать на всех пользователей машины либо только на текущего пользователя.

Метод .Delete() (удалить)

Метод для удаления объекта XMLNamespace из коллекции и уничтожения его, эффективно удаляющий связь пространства имен, представленную этим объектом, из библиотеки пространств имен.

Свойство .Location (расположение)

Доступное только на чтение свойство, которое контролирует расположение схемы, связанной с пространством имен, представленным упомянутым объектом XMLNamespace. Оно может поддерживать следующий параметр.

AllUsers (все пользователи) - флаг, показывающий, должна ли установка расположения схемы воздействовать на всех пользователей машины либо только на текущего пользователя.

Свойство .Parent (родитель)

Доступное только на чтение свойство, возвращающее родительский объект объекта XMLNamespace. Это свойство возвращает указатель на коллекцию XMLNamespaces, членом которой является данный объект XMLNamespace.

Свойство .URI (универсальный идентификатор ресурса)

Доступное только на чтение свойство, возвращающее URI пространства имен, представленного объектом.

Свойство .XSLTransforms (XSLT-преобразования)

Доступный только на чтение указатель на коллекцию XSLTransforms XSLT-преобразований, представляющую XSLT-преобразования, связанные с пространством имен, представленным упомянутым объектом.

Объект XSLTransforms (XSLT-преобразования) - объект, обеспечивающий доступ к объектам XSLTransforms, каждый из которых представляет одиночное и уникальное XSLT-преобразование, связанное с пространством имен в библиотеке пространств имен. Ниже приведены методы и свойства такого объекта.

Метод .Add() (добавить)

Метод для создания и добавления к коллекции нового объекта XSLTransform. Он используется для связывания нового XSLT-преобразования с пространством имен в библиотеке пространств имен. Он возвращает новый объект XSLTransform. Он может принимать следующие параметры.

Location - указатель на XSLT-файл; может быть путем файла, представленным как текстовая строка.

Alias - текстовая строка, представляющая альтернативное (более дружественное пользователю) имя для XSLT-преобразования, которое может быть задано программистом.

InstallForAllUsers - флаг, показывающий, должно ли упомянутое новое XSLT-преобразование в библиотеке пространств имен быть доступно всем пользователям компьютера либо только текущему пользователю.

Свойство .Application (приложение)

Доступный только на чтение указатель на объект Application, представляющий приложение этой объектной модели.

Свойство .Count (счет)

Доступное только на чтение свойство, возвращающее количество

зарегистрированных XSLT-преобразований для заданного пространства имен в библиотеке пространств имен. Это свойство идентично общему количеству объектов XSLTransform в коллекции XSLTransforms.

Свойство .Creator (создатель)

Доступный только на чтение указатель на создателя этого объекта.

Метод .Item() (Получить компоненту)

Метод для осуществления доступа к отдельным членам этой коллекции, использующий числовой индекс или ключевое слово поиска. Такой метод может принимать следующие параметры.

Index (индекс) - номер, представляющий расположение запрашиваемого объекта XSLTransform в библиотеке пространств имен. Такой индекс может также быть текстовой строкой, представляющей псевдоним запрашиваемого XSL-преобразования.

Свойство .Parent (родитель)

Доступное только на чтение свойство, возвращающее родительский объект такой коллекции. Данное свойство возвращает указатель на приложение, из которого осуществлен доступ к коллекции XSLTransforms.

Объект XSLTransform (XSLT-преобразование) - объект, представляющий объект XSLT-преобразования, связанный с пространством имен в библиотеке пространств имен. Ниже приведены методы и свойства такого объекта.

Свойство .Alias (псевдоним)

Свойство для управления псевдонимом, связанным программистом с XSLT-преобразованием в библиотеке пространств имен. Оно может поддерживать следующий параметр.

AllUsers (все пользователи) - флаг, показывающий, доступен ли псевдоним всем пользователям либо текущему пользователю.

Свойство .Application (приложение)

Доступный только на чтение указатель на объект Application, представляющий приложение этой объектной модели.

Свойство .Creator (создатель)

Доступный только на чтение указатель на создателя этого объекта.

Метод .Delete() (удалить)

Метод для удаления объекта из коллекции и уничтожения его, эффективно удаляющий связь между объектом XSLTransform и его пространством имен в библиотеке пространств имен.

Свойство .Location (расположение)

Доступное только на чтение свойство, которое контролирует расположение XSLT-преобразования, связанного с данным пространством имен и представленного объектом XSLTransform. Оно может поддерживать следующий параметр.

AllUsers (все пользователи) - флаг, показывающий, должна ли установка расположения XSLT-преобразования воздействовать на всех пользователей машины либо только на текущего пользователя.

Свойство .Parent (родитель)

Доступное только на чтение свойство, возвращающее родительский объект упомянутого объекта XSLTransform. то свойство возвращает указатель на коллекцию XSLTransforms, членом которого является данный объект.

В соответствии с приведенным описанием способы и система предоставлены для обеспечения пользователю возможности программным путем осуществлять вызов

ресурсов, идентифицированных в библиотеке пространства имен или схем расширяемого языка разметки для настройки или иного модифицирования связи ресурсов, идентифицированных или содержащихся в библиотеке пространств имен или схем, с одним или более ассоциированных документов. Для специалистов в данной области техники будет очевидным, что разнообразные модификации или вариации могут быть сделаны в настоящем изобретении без отклонения от объема и сущности этого изобретения. Другие воплощения данного изобретения будут очевидны специалистам в данной области техники из анализа описания и практического применения раскрытого здесь изобретения.

Формула изобретения

1. Компьютерно-реализованный способ изменения ресурсов библиотеки схем языка разметки (ML), содержащий этапы, на которых

вызывают из средства редактирования документов библиотеку ML-схем через объектно-ориентированный привязанный к сообщению вызов, причем этот объектно-ориентированный привязанный к сообщению вызов приспособлен для доступа к и изменения содержимого файла ML-схемы библиотеки ML-схем, применяемой к документу;

исполняют метод объектно-ориентированного привязанного к сообщению вызова для

создания нового пространства имен ML,

добавления этого нового пространства имен ML к совокупности пространств имен ML, причем путь к файлу схемы, связанному с новым пространством имен ML, и универсальный указатель ресурса для нового пространства имен ML передаются в библиотеку ML-схем в качестве параметров упомянутого метода,

установки манифестов решений для регистрации пространств имен ML в

библиотеке ML-схем и

присоединения файла схемы пространства имен ML к документу, при этом указатель на данный документ передается в библиотеку ML-схем в качестве параметра упомянутого метода;

в качестве реакции на исполнение упомянутого метода в отношении

библиотеки ML-схем обуславливают изменение содержимого по меньшей мере одного пространства имен файла схемы и

применяют это изменение файла ML-схемы к документу для реализации применения ML-элементов к документу.

2. Способ по п.1, в котором при исполнении метода исполняют метод для осуществления доступа к отдельным ML-ресурсам из совокупности ML-ресурсов, используя числовой индекс, при этом числовой индекс, связанный с отдельным ML-ресурсом, передается в качестве параметра упомянутого метода.

3. Способ по п.1, в котором при исполнении метода исполняют метод для управления псевдонимом, связанным с заданным пространством имен, идентифицированным в библиотеке ML-схем.

4. Способ по п.1, дополнительно содержащий этап, на котором передают свойство объекта, которое указывает на применяемое по умолчанию преобразование расширяемого языка таблиц стилей (XSLT), связанное с заданным пространством имен.

5. Способ по п.1, в котором при исполнении метода исполняют метод для удаления объекта пространства имен ML из совокупности объектов пространств имен.

6. Способ по п.1, в котором при исполнении метода исполняют метод для создания нового XSLT-преобразования и для добавления этого нового XSLT-преобразования к совокупности XSLT-преобразований, причем указатель на это новое XSLT-преобразование передается в библиотеку ML-схем в качестве параметра

7. Способ по п.1, в котором при исполнении метода исполняют метод для осуществления доступа к отдельным XSLT-преобразованиям, содержащимся в совокупности XSLT-преобразований, используя числовой индекс, при этом числовой индекс, представляющий расположение запрашиваемого XSLT-преобразования в библиотеке ML-схем, передается в библиотеку ML-схем в качестве параметра упомянутого метода.

8. Способ по п.1, в котором при исполнении метода исполняют метод для управления псевдонимом, связанным с XSLT-преобразованием, идентифицированным в библиотеке ML-схем.

9. Способ по п.1, в котором при исполнении метода исполняют метод для удаления XSLT-преобразования из совокупности XSLT-преобразований.

10. Считываемый компьютером носитель, на котором имеются исполняемые компьютером команды для изменения ресурсов библиотеки схем языка разметки (ML), содержащие

прием объектно-ориентированного привязанного к сообщению вызова в отношении библиотеки ML-схем, причем этот объектно-ориентированный привязанный к сообщению вызов приспособлен для доступа к и изменения содержимого файла ML-схемы библиотеки ML-схем;

исполнение метода объектно-ориентированного привязанного к сообщению вызова для

добавления нового пространства имен ML к совокупности пространств имен ML, причем путь к файлу схемы, связанному с новым пространством имен ML, и универсальный указатель ресурса для нового пространства имен ML передаются в библиотеку ML-схем в качестве параметров упомянутого метода,

установки манифестов решений для регистрации пространств имен ML в библиотеке ML-схем и

присоединения файла схемы пространства имен ML к документу, при этом указатель на данный документ передается в библиотеку ML-схем в качестве параметра упомянутого метода;

приведение в качестве реакции на исполнение упомянутого метода в отношении библиотеки ML-схем к изменению содержимого по меньшей мере одного пространства имен файла схемы и

ассоциирование этого изменения файла ML-схемы с документом для реализации применения ML-элементов к документу.

11. Считываемый компьютером носитель по п.10, в котором исполнение упомянутого метода в отношении библиотеки ML-схем приводит к удалению пространства имен из файла схемы.

12. Считываемый компьютером носитель по п.10, в котором исполнение упомянутого метода в отношении библиотеки ML-схем приводит к по меньшей мере одному из ассоциирования XSLT-преобразования с файлом схемы и удаления ассоциативной связи XSLT-преобразования с файлом схемы.

13. Считываемый компьютером носитель по п.10, в котором исполнение упомянутого метода в отношении библиотеки ML-схем приводит к по меньшей мере

одному из ассоциирования основывающегося на ML ресурса с файлом схемы и удаления ассоциативной связи ML ресурса с файлом схемы.

14. Компьютерная система для изменения ресурсов библиотеки схем языка разметки (ML), содержащая

процессор;

память, в которой хранятся машиноисполняемые команды, приспособленные для приема объектно-ориентированного привязанного к сообщению вызова в отношении библиотеки ML-схем, причем этот объектно-ориентированный привязанный к сообщению вызов приспособлен для доступа к и изменения содержимого файла ML-схемы библиотеки ML-схем;

исполнения метода объектно-ориентированного привязанного к сообщению вызова для:

добавления нового пространства имен ML к совокупности пространств имен ML, причем путь к файлу схемы, связанному с новым пространством имен ML, и универсальный указатель ресурса для нового пространства имен ML передаются в библиотеку ML-схем в качестве параметров упомянутого метода,

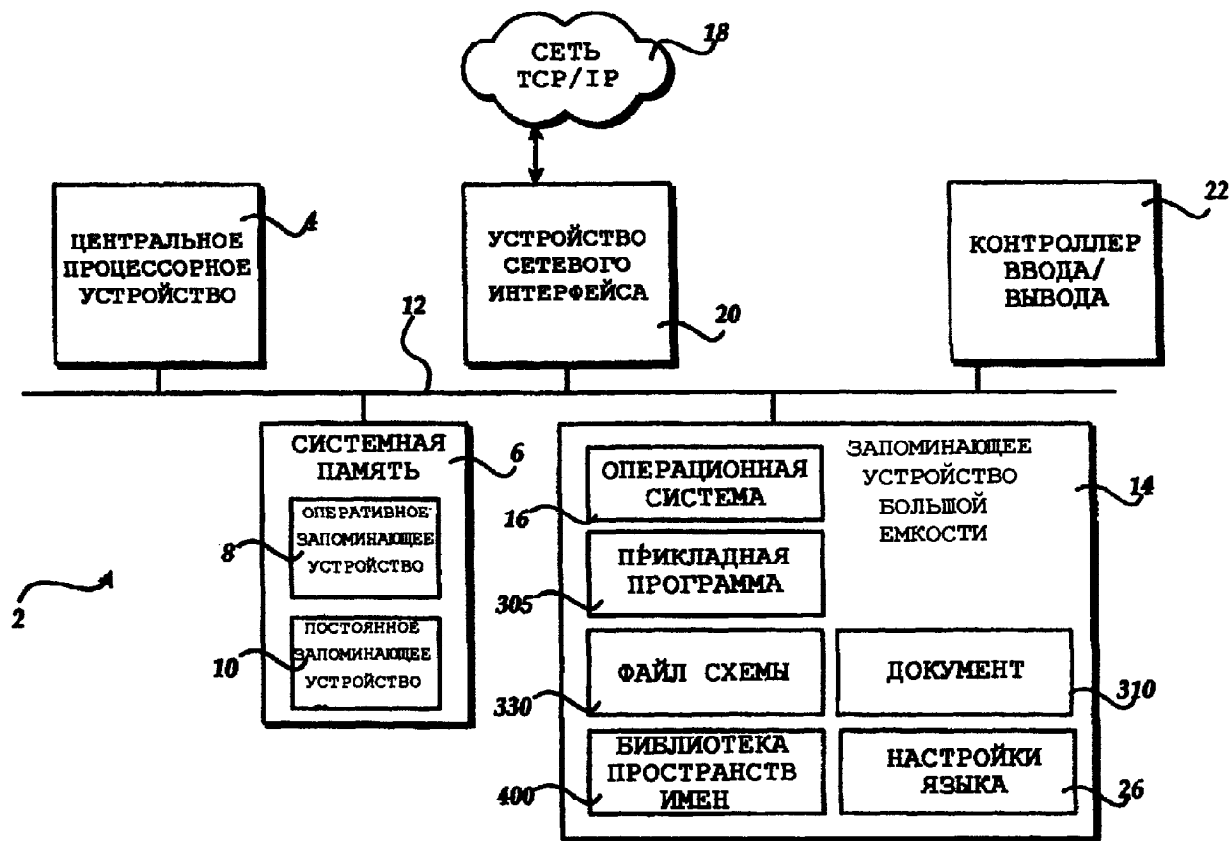
установки манифестов решений для регистрации пространств имен ML в библиотеке ML-схем и

присоединения файла схемы пространства имен ML к документу, при этом указатель на данный документ передается в библиотеку ML-схем в качестве параметра упомянутого метода;

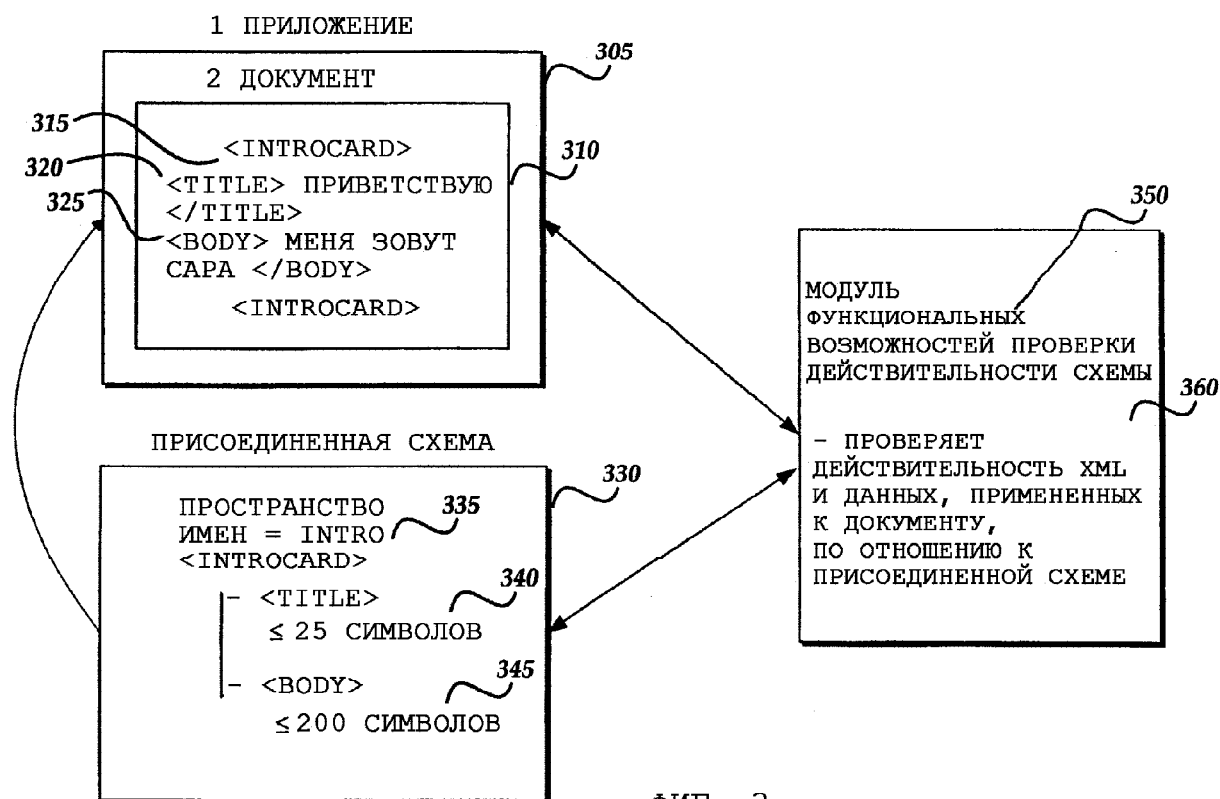
приведения в качестве реакции на исполнение упомянутого метода в отношении библиотеки ML-схем к изменению содержимого по меньшей мере одного пространства имен файла схемы и

ассоциирования этого изменения файла ML-схемы с документом для реализации применения ML-элементов к документу.

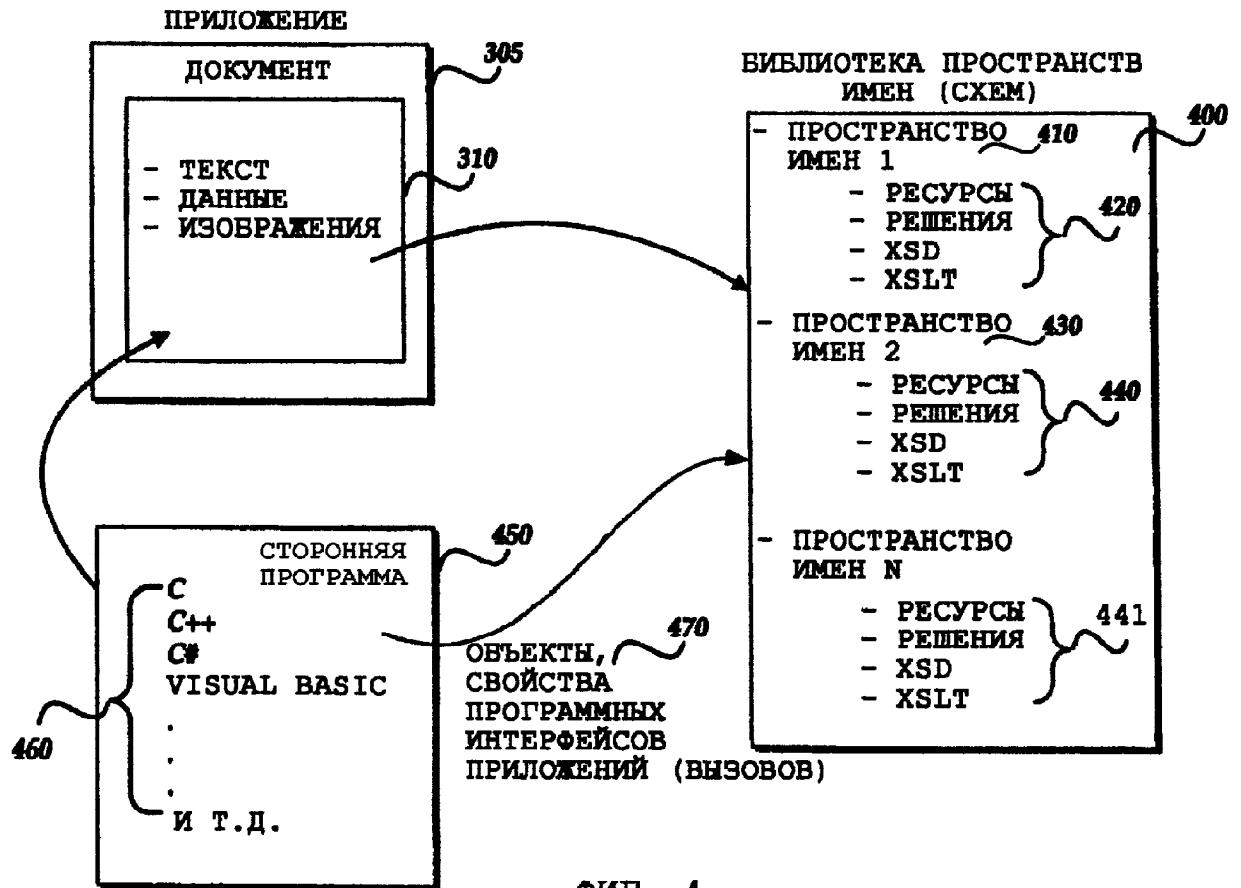
15. Система по п.14, в которой исполнение упомянутого метода в отношении библиотеки ML-схем приводит к одному из удаления пространства имен из файла схемы, ассоциирования XSLT-преобразования с файлом схемы, удаления ассоциативной связи XSLT-преобразования с файлом схемы, ассоциирования основывающегося на ML ресурса с файлом схемы и удаления ассоциативной связи ML ресурса с файлом схемы.



ФИГ. 1



ФИГ. 3



ФИГ. 4