



(51) МПК

G06F 9/44 (2006.01)*G06F 17/24* (2006.01)*G06F 9/46* (2006.01)*G06N 1/00* (2006.01)

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ,
ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21), (22) Заявка: 2004105880/09, 27.02.2004

(24) Дата начала отсчета срока действия патента:
27.02.2004(30) Конвенционный приоритет:
28.02.2003 US 10/377,258

(43) Дата публикации заявки: 10.08.2005

(45) Опубликовано: 27.02.2009 Бюл. № 6

(56) Список документов, цитированных в отчете о поиске: US 5765156 A, 09.06.1998. "Adding data from other sources in PowerPoint 2002&", опубликованная в июле 2002, найденная на сайте.
<http://www.glencoe.com/ps/computered/pas/article.php4?articleId=437>, в разделе "A professional approach series' archives", с.1-4. FRANK C.RICE «Transform XML file with XSLT when importing into Microsoft Excel 2002», (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):

ДЖОУНС Брайан Майкл (US),
САВИЦКИ Марсин (US),
ЛИТТЛ Роберт А. (US),
МЮРРЭЙ Майкл (US),
КОЛЛИ Роберт (US)

(73) Патентообладатель(и):

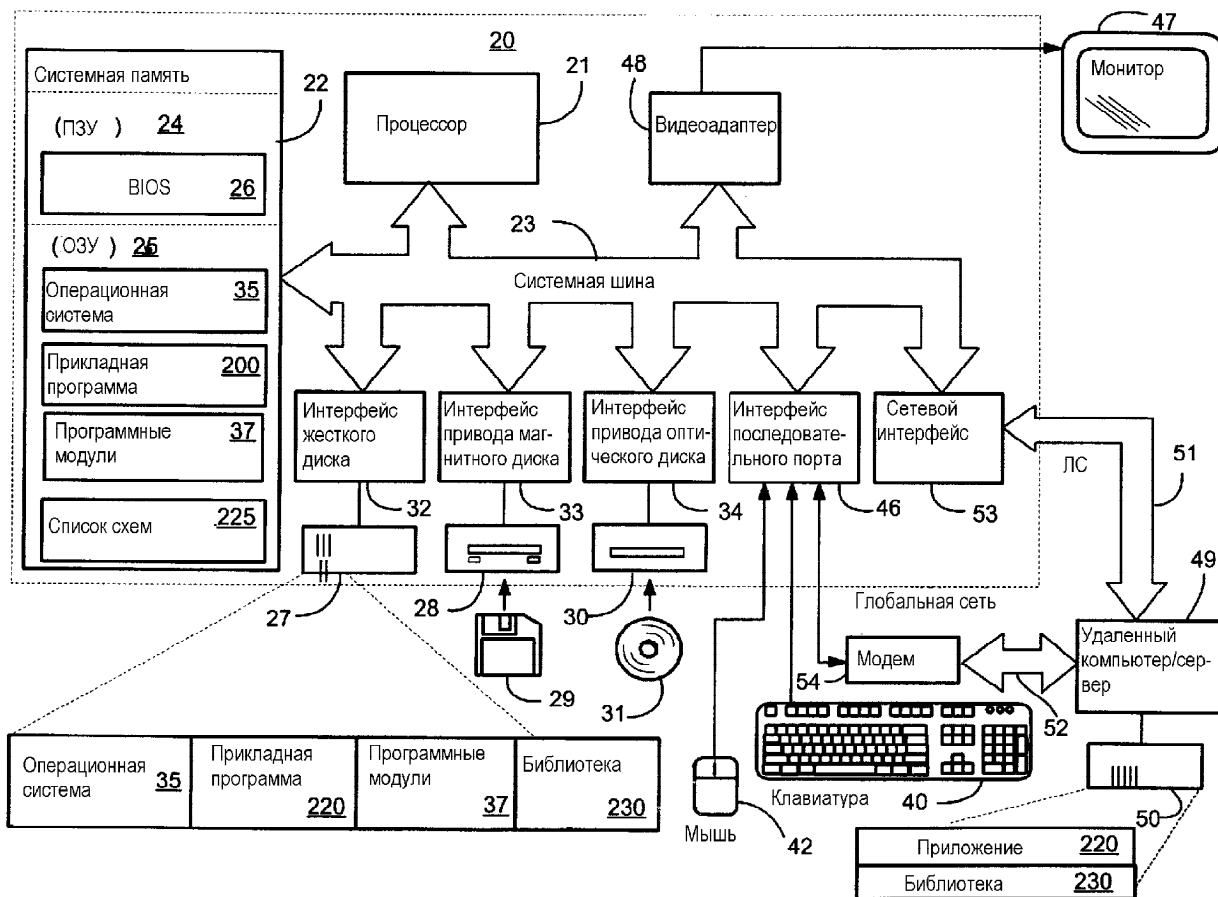
МАЙКРОСОФТ КОРПОРЕЙШН (US)

(54) СПОСОБ И СИСТЕМА ДЛЯ РАСШИРЕНИЯ ФУНКЦИОНАЛЬНЫХ ВОЗМОЖНОСТЕЙ ВСТАВКИ ДЛЯ КОМПЬЮТЕРНОГО ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

(57) Реферат:

Изобретение относится к вычислительной технике. Техническим результатом является устранение несовместимости при вставке. Данные выбирают из документа, чтобы вставить их во второй документ. Выбранные данные сохраняются в памяти, например в буфере обмена. Совместно с выбранными данными сохраняется информация или данные, связанные с выбранными данными, для предоставления прикладной программе, осуществляющей вставку, или приложению-потребителю информации о любых типах данных,

связанных с выбранными данными. Соответственно, приложение-потребитель может получить пространство имен и соответствующий ресурс, например файл преобразования расширяемого языка стилей, для преобразования выбранных данных из первого типа данных, связанного с первым документом, во второй тип данных для вставки выбранных данных во второй документ, чтобы сохранить во вставленных данных структуру и форматирование данных, которые применялись к выбранным данным до вставки. 4 н. и 29 з.п. ф-лы, 6 ил.



Фиг. 1

(56) (продолжение):

опубликованная в июле 2001, найденная на сайте [http://msdn2.microsoft.com/en-us/library/aal40055\(printer\).aspx](http://msdn2.microsoft.com/en-us/library/aal40055(printer).aspx), разделы «Introduction» - «Creating an XML Data file», «Importing data by using a style sheet» - «An example of an XSLT style sheet». US 5386564 A, 31.01.1995. ДЖЕРАЛЬ Р. РОУТЛЕДЖ и др. Ваш персональный компьютер, Москва, Бином, 1995, с.203-223, 234-244.

RU 2348064 C2

RU 2348064 C2



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY,
PATENTS AND TRADEMARKS

(51) Int. Cl.
G06F 9/44 (2006.01)
G06F 17/24 (2006.01)
G06F 9/46 (2006.01)
G06N 1/00 (2006.01)

(12) **ABSTRACT OF INVENTION**

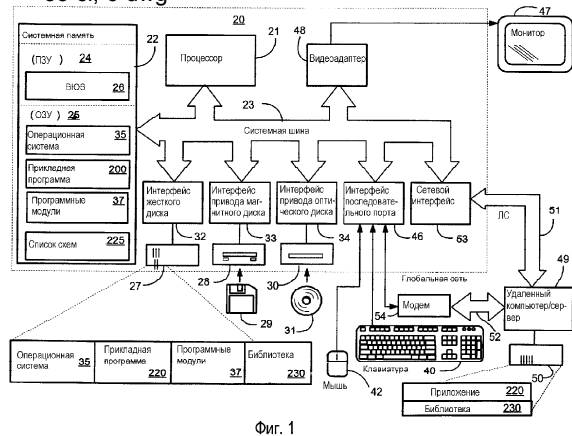
(21), (22) Application: **2004105880/09, 27.02.2004**
(24) Effective date for property rights: **27.02.2004**
(30) Priority:
28.02.2003 US 10/377,258
(43) Application published: **10.08.2005**
(45) Date of publication: **27.02.2009 Bull. 6**
Mail address:
129090, Moskva, ul. B.Spasskaja, 25, str.3,
OOO "Juridicheskaja firma Gorodisskij i
Partnery", pat.pov. Ju.D.Kuznetsovu, reg.№ 595

(72) Inventor(s):
DZhOUNS Brajan Majkl (US),
SAVITSKI Marsin (US),
LITTL Robert A. (US),
MJuRREHJ Majkl (US),
KOLLI Robert (US)
(73) Proprietor(s):
MAJKROSOFT KORPOREJShN (US)

(54) **METHOD AND SYSTEM OF EXTENDING FUNCTIONAL CAPACITY OF INSERTION FOR COMPUTER SOFTWARE APPLICATIONS**

(57) Abstract:
FIELD: physics, computer technology.
SUBSTANCE: invention concerns computer technology. Data for insertion to the second document are selected in the first document. Selected data are stored in memory, such as clipboard. Together with selected data, related information or data is copied for presentation of any data types related to selected data to application performing insertion or to data user application. User application can respectively obtain name space and related resource, e.g. extensible style language transformation file, for transformation of selected data from first data type related to the first document to another data type for insertion of selected data to the second document with preservation of data structure and format applied in selected data

before insertion in inserted data.
EFFECT: prevention of incompatibility of insertion.
33 cl, 6 dwg



RU 2 348 064 C2

RU 2 348 064 C2

Область техники, к которой относится изобретение

Данное изобретение относится к способам и системам для расширения функциональных возможностей вставки, доступных компьютерному прикладному программному обеспечению для вставки данных в документ, создаваемом на компьютере.

5 Предшествующий уровень техники

Прикладные компьютерные программы позволяют пользователю создавать различные документы для работы, образования и досуга. Например, приложения текстового редактора позволяют пользователю создавать письма, статьи, книги, заметки и т.п. Приложения электронных таблиц позволяют пользователю сохранять, обрабатывать, распечатывать и
10 отображать различные буквенно-цифровые данные. Такие приложения имеют ряд общеизвестных функциональных особенностей, включая развитые возможности редактирования, форматирования, печати и вычислений.

Многие прикладные программы снабжены общеизвестной и весьма полезной функцией, позволяющей вырезать или копировать данные из заданного документа, электронной
15 таблицы, слайдовой презентации или другого документа, создаваемого на компьютере, чтобы затем можно было выполнять операцию вставки, позволяющую вставлять вырезанные или скопированные данные в документ в нужном месте. Зачастую пользователь вырезает или копирует данные, в том числе текст, изображения или буквенно-цифровые данные из первого приложения или «приложения-поставщика»,
20 например приложения текстового редактора, а затем вставляет вырезанные или скопированные данные с использованием второго приложения или приложения-потребителя, например приложения электронной таблицы. Например, пользователю может потребоваться вырезать или скопировать массив чисел из документа приложения электронной таблицы, чтобы вставить эти числа в документ приложения текстового
25 редактора, в котором пользователь составляет письмо для отправки клиенту или сотруднику. Обычно при вырезании или копировании данных данные буферизуются в памяти для последующей вставки при выполнении операции вставки.

При вырезании или копировании данных из приложения-поставщика с последующей вставкой в другой документ посредством второго приложения, «приложения-потребителя»,
30 причем приложение-поставщик и приложение-потребитель являются разными прикладными программами, часто случается так, что многие, если не все, особенности, обеспечиваемые первым приложением, включая форматирование, утрачиваются при выполнении операции вставки, поскольку второе приложение или приложение-потребитель не содержит функций, необходимых для обеспечения специальных особенностей, включая
35 форматирование, поддерживаемых первым приложением. Например, если пользователь копирует массив данных из документа приложения электронной таблицы, а затем вставляет эти данные в документ программы текстового редактора, то данные, вставленные в документ программы текстового редактора, могут оказаться лишенными какого-либо форматирования, имевшего место в приложении электронной таблицы.
40 Например, если данные в приложении электронной таблицы были представлены в нескольких столбцах и строках, то, будучи вставлены в приложение текстового редактора, эти данные могут иметь вид простой последовательности чисел без какого бы то ни было форматирования.

Чтобы снабдить прикладные программы более развитыми функциями, разработчики
45 программного обеспечения начали использовать языки разметки, например расширяемый язык разметки (XML), позволяющий пользователю аннотировать документ прикладной программы, чтобы придавать документу полезную структуру, помимо обычных функций прикладной программы, отвечающих за создание документа или отображаемое форматирование, связанное с документом. Например, пользователь может пожелать
50 создать в своем приложении текстового редактора шаблон документа для подготовки статьи, которую он намерен отослать издателю. Благодаря применению структуры к документу издатель, получив документ, может использовать эту структуру при обработке документа, чтобы использовать данные, указанные в структуре документа. К сожалению,

при вырезании или копировании данных из документа, имеющего такую структурную аннотацию, структурная аннотация часто утрачивается в операции вставки, особенно, когда операция вставки осуществляется другим приложением-потребителем. Даже если приложение-потребитель запрограммировано понимать и использовать язык разметки, 5
схема или правила, задающие структурную аннотацию языка разметки, могут значительно отличаться для приложения-поставщика и приложения-потребителя.

Эти и другие соображения привели к созданию настоящего изобретения.

Сущность изобретения

Настоящее изобретение обеспечивает способы и системы для расширения 10
функциональных возможностей вставки, доступных прикладному компьютерному программному обеспечению для вставки данных в документ, создаваемый на компьютере. В общем случае, данные выбирают из первого приложения, чтобы вставить в документ второго приложения. Выбранные данные совместно с информацией, связанной с данными, например список пространств имен, сохраняются в памяти. При вставке выбранных данных 15
в документ второго приложения или приложения-потребителя приложение-потребитель проверяет доступный список пространств имен, связанных с выбранными данными, чтобы определить, понятны ли различные представления данных приложению, осуществляющему вставку. Приложение, осуществляющее вставку, может также осуществлять поиск доступных ресурсов, например файлов преобразования расширяемого языка стилей (XSL) для преобразования выбранных данных в формат, легче 20
воспринимаемый вторым приложением или приложением-потребителем. Один способ нахождения ресурсов для помощи с доступными пространствами имен состоит в том, что приложение-потребитель проверяет библиотеку пространств имен на предмет наличия ресурса для преобразования данных для использования приложением-потребителем. Если 25
никаких ресурсов не найдено, и приложение, осуществляющее вставку, по природе своей, не понимает пространств имен, то вставка данных в документ второго приложения может производиться согласно ресурсу вставки, заданному по умолчанию. В приложении, осуществляющем вставку, также может быть предусмотрен способ, заданный по умолчанию, позволяющий работать с любым типом данных независимо от пространства 30
имен.

В частности, предусмотрен способ расширения функциональных возможностей вставки для прикладного компьютерного программного обеспечения. Открывают первый документ через приложение-поставщик и применяют структуру к первому документу согласно языку разметки, например расширяемому языку разметки (XML). Возможен случай, когда первое 35
приложение служит только для визуального отображения существующих данных. Данные выбирают из первого приложения для вставки во второе приложение. Выбранные данные сохраняются в памяти, и информация, связанная с выбранными данными, сохраняется в памяти для предоставления приложению-потребителю информации о выбранных данных. Эта информация обычно представляет собой список пространств имен. Эти пространства 40
имен указывают разные способы представления данных в приложении, осуществляющем копирование. Рассмотрим, например, документ электронной таблицы, содержащий финансовые данные. Можно обеспечить одно пространство имен для описания электронной таблицы, а другое - для описания финансовых данных.

Открывают второй документ через приложение-потребитель и выбирают в приложении- 45
потребителе функцию вставки для вставки выбранных данных во второй документ. Приложение-потребитель считывает информацию, связанную с выбранными данными. В соответствии с ней приложение-потребитель определяет, может ли приложение-потребитель по своей сути понимать одно или несколько пространств имен, идентифицирующих типы данных, связанные с выбранными данными. При наличии 50
нескольких пространств имен приложение может либо предоставлять пользователю возможность выбора пространства имен и соответствующих ресурсов для использования, либо самостоятельно выбирать пространство имен и соответствующие ресурсы, которые наилучшим образом подходят к данным, содержащимся в документе приложения-

потребителя. Представленные варианты выбора могут позволять выбирать используемое пространство имен, а также могут позволять выбирать используемый ресурс для любого заданного пространства имен. Если приложение-потребитель не понимает этого пространства имен, то приложение-потребитель может получить один или несколько файлов ресурсов, например файлы преобразования XSLT, связанные с пространствами имен, для вставки выбранных данных во второй документ. После того, как приложение-потребитель выберет одно или несколько пространств имен для вставки выбранных данных во второй документ, выбранные данные вставляются во второй документ согласно способу, предпочтительному для приложения-потребителя. Иногда используемый дополнительный ресурс может представлять собой файл преобразований расширенного языка стилей (XSLT) для преобразования выделенных данных к типу или формату данных, воспринимаемому вторым приложением или приложением-потребителем.

Согласно одному аспекту изобретения до получения ресурса для одного или нескольких пространств имен для вставки выбранных данных производится определение, содержит ли список доступных пространств имен нужное пространство имен из одного или нескольких пространств имен, понимаемых приложением-потребителем. Если список доступных пространств имен содержит нужное пространство имен из одного или нескольких пространств имен, понимаемых приложением-потребителем, то нужное пространство имен выбирают и предоставляют приложению-потребителю. Если в списке доступных пространств имен нет нужного пространства имен из одного или нескольких пространств имен, понимаемых приложением-потребителем, то производится определение, содержит ли библиотека пространств имен ресурс, помогающий приложению-потребителю понимать одно или несколько пространств имен для использования приложением-потребителем. Если библиотека пространств имен содержит ресурс для одного или нескольких пространств имен для использования приложением-потребителем, то нужный ресурс выбирают из библиотеки пространств имен и предоставляют приложению-потребителю, чтобы помочь ему потреблять одно или несколько пространств имен.

Информация, связанная с выбранными данными, сохраняется в памяти для предоставления приложению-потребителю и может содержать индикацию наличия одного или нескольких пространств имен, связанных с выбранными данными. Для каждого пространства имен, связанного с выбранными данными, информация, связанная с выбранными данными, может содержать информацию о любом соответствующем пространстве имен, включающую в себя идентификатор версии для пространства имен, универсальный идентификатор ресурса для размещения пространства имен и размер файла для пространства имен.

Согласно другому аспекту изобретения обеспечен способ вставки данных из приложения, осуществляющего копирование, в приложение-потребитель. Первый элемент расширяемого языка разметки (XML) применяют к началу области первого документа. Область выделяют для копирования из приложения, осуществляющего копирование, и для вставки в приложение-потребитель. Второй элемент XML применяют к концу выбранной области. Информацию обеспечивают в заголовке, связанном с выбранной областью. Информация содержит идентификационные данные доступных пространств имен XML и указатели на идентификационные данные формата буфера обмена, который соответствует каждому из доступных пространств имен. В заголовке обеспечивают информацию о размере файла для каждого из доступных пространств имен. Одно или несколько пространств имен из доступных пространств имен обеспечивают в буфере обмена, чтобы приложение-потребитель могло выбирать из одного или нескольких пространств имен для вставки данных из приложения, осуществляющего копирование.

Эти и другие признаки и преимущества настоящего изобретения явствуют из нижеследующего подробного описания, приведенного со ссылками на прилагаемые чертежи. Следует понимать, что как вышеизложенное общее описание, так и нижеследующее подробное описание носят исключительно иллюстративный и пояснительный характер и не ограничивают заявленное изобретение.

Перечень фигур чертежей

Фиг.1 - блок-схема компьютера и соответствующих ему периферийных и сетевых устройств, которые обеспечивают иллюстративную операционную среду для настоящего изобретения.

5 Фиг.2 - упрощенная блок-схема, иллюстрирующая взаимодействие между приложением-поставщиком и приложением потребителем при вырезании или копировании данных с использованием приложения-поставщика и вставке с использованием приложения-потребителя.

10 Фиг.3 - упрощенная блок-схема, иллюстрирующая пользовательские интерфейсы для предоставления пользователям расширенных функций вставки согласно иллюстративному варианту осуществления настоящего изобретения.

Фиг.4 - отображение на экране компьютера прикладной программы, вставляющей и потребляющей данные, вырезанные или скопированные из приложения-поставщика.

15 Фиг.5 и 6 - логические блок-схемы способа расширения функций вставки приложения-потребителя согласно варианту осуществления настоящего изобретения.

Подробное описание предпочтительного варианта осуществления

20 Подробное описание вариантов осуществления настоящего изобретения приведено ниже со ссылкой на вышеперечисленные чертежи, на которых сходные позиции обозначают аналогичные части или компоненты на нескольких фигурах. Настоящее изобретение относится к способам и системам для расширения функций вставки, доступных прикладному компьютерному программному приложению для вставки данных в документ, создаваемый на компьютере. Пользователь выбирает данные из первого приложения, например приложения электронной таблицы, для вставки во второе приложение, например приложение текстового редактора. Процесс вставки может 25 происходить в виде традиционной операции вырезания/копирования и вставки, а может происходить в виде перетаскивания и сброса выделенных данных из документа первого приложения в документ второго приложения. Согласно варианту осуществления настоящего изобретения первое приложение или приложение-поставщик записывает выбранные данные в представлении расширяемого языка разметки (XML) в память, 30 например буфер обмена. Выбранные данные записываются в память в формате, который включает в себя выбранные данные и информацию, связанную с одним или несколькими пространствами имен данных, связанными с выбранными данными. Доступные пространства имен, связанные с выбранными данными, могут идентифицировать типы форматирования данных, которые могут быть связаны с выбранными данными. Например, 35 идентифицированные доступные пространства имен могут включать в себя язык гипертекстовой разметки (HTML), документ резюме или XML текстового редактора. Пространство имен HTML может идентифицировать, что данные могут иметь структуру HTML. Пространство имен документа резюме может идентифицировать, что данные могут быть структурированы согласно файлу схемы XML резюме. Пространство имен XML текстового редактора может идентифицировать, что данные могут быть форматированы 40 согласно версии текстового редактора для XML. Конечно, эти пространства имен являются лишь примером многочисленных типов пространств имен, которые могут быть связаны с выбранными данными.

45 При вставке выбранных данных в документ второго приложения путем выбора функции вставки второго приложения или путем сброса выбранных данных в документ второго приложения второе приложение или приложение-потребитель обнаруживает из информации, предоставленной совместно с выбранными данными, что вставляемые данные имеют формат XML. На основании информации, предоставленной совместно с 50 выбранными данными, приложение-потребитель обнаруживает все доступные пространства имен, связанные со вставляемыми данными. Например, в вышеприведенном примере приложение-потребитель может обнаружить, что с выбранными данными связаны пространства имен для HTML, документа резюме и XML текстового редактора. Соответственно, приложение-потребитель ищет в списке доступных пространств имен

ресурс, помогающий приложению-потребителю вставить выделенные данные. Например, если приложение-поставщик является приложением электронной таблицы, то приложение-потребитель может искать в списке доступных пространств имен файл преобразования расширяемого языка стилей (XSLT) для преобразования данных в формате XML

5 электронной таблицы в формат для потребления и использования приложением-потребителем. Если приложение-потребитель является текстовым редактором, то приложение-потребитель может искать файл преобразования XSLT для преобразования данных электронной таблицы в данные для потребления приложением текстового редактора. Например, если данные были структурированы в приложении электронной
10 таблицы в формате, содержащем две строки и три столбца, то файл преобразования XSLT, используемый приложением-потребителем, может преобразовывать эти данные для потребления приложением текстового редактора таким образом, чтобы данные остались в формате двух строк и трех столбцов. В отсутствие функций настоящего изобретения данные могут быть вставлены согласно функциям вставки приложения-потребителя,
15 принятым по умолчанию, и, таким образом, могут быть вставлены без форматирования приложения-поставщика. Следовательно, в описываемом примере данные могут быть вставлены в документ приложения текстового редактора в качестве необработанных данных, представленных слева направо через рабочее пространство текстового редактора, и не будут форматированы в виде двух строк и трех столбцов.

20 Если для приложения-потребителя идентифицированы несколько доступных пространств имен, то приложение-потребитель может выбрать доступное пространство имен, которое с наибольшей вероятностью обеспечит приложение-потребитель нужными функциями вставки. Альтернативно пользователю приложения-потребителя может быть предоставлен пользовательский интерфейс, позволяющий пользователю выбирать из
25 доступных файлов преобразования, известных приложению-потребителю, для списка доступных пространств имен. Если приложение-потребитель не нашло в списке доступных пространств имен никаких ресурсов, то приложение-потребитель может перейти к библиотеке пространств имен, чтобы искать доступный ресурс, помогающий приложению-потребителю преобразовать выбранные данные, до вставки данных в документ
30 приложения-потребителя. Если в библиотеке пространств имен имеется несколько ресурсов, то пользователю может быть предоставлен пользовательский интерфейс, позволяющий выбирать из доступных ресурсов. Если никаких доступных ресурсов нет ни в списке доступных пространств имен, ни в библиотеке пространств имен, то приложение-потребитель вставляет данные согласно функциям, принятым по умолчанию, как описано
35 выше.

Операционная среда

Ниже, со ссылкой на фиг.1, приведено краткое общее описание подходящей вычислительной среды, в которой можно реализовать изобретение. Хотя изобретение описано в общем контексте прикладной программы, которая работает под операционной
40 системой персонального компьютера, специалистам в данной области очевидно, что изобретение можно также реализовать в сочетании с другими программными модулями. Обычно программные модули включают в себя процедуры, программы, компоненты, структуры данных и т.д., предназначенные для решения конкретных задач или реализации тех или иных абстрактных типов данных. Кроме того, специалистам в данной области
45 очевидно, что изобретение можно осуществлять на практике в компьютерных системах других конфигураций, включая карманные устройства, многопроцессорные системы, бытовую электронику на основе микропроцессора или программируемую бытовую электронику, сотовые телефоны, миникомпьютеры, универсальные компьютеры и т.п. Изобретение также можно осуществлять на практике в распределенных вычислительных
50 средах, где задачи решаются удаленными обрабатывающими устройствами, соединенными посредством сети связи. В распределенной вычислительной среде программные модули могут располагаться как в локальных, так и в удаленных запоминающих устройствах.

Согласно фиг.1 иллюстративная система для реализации изобретения включает в себя традиционный персональный компьютер 20, содержащий процессор 21, системную память 22 и системную шину 23, которая подключает системную память к процессору 21.

5 Системная память 22 включает в себя постоянную память (ПЗУ) 24 и оперативную память (ОЗУ) 25. Базовая система ввода/вывода 26 (BIOS), содержащая основные процедуры, помогающие переносить информацию между элементами персонального компьютера 20, например, при запуске, хранится в ПЗУ 24. Персональный компьютер 20 также содержит жесткий диск 27, привод 28 магнитного диска, например, для чтения или записи сменного диска 29, и привод 30 оптического диска, например, для чтения диска 31 CD-ROM или
10 чтения или записи другого оптического носителя. Жесткий диск 27, привод 28 магнитного диска и привод 30 оптического диска подключены к системной шине 23 посредством интерфейса 32 жесткого диска, интерфейса 33 привода магнитного диска и интерфейса 34 привода оптического диска соответственно. Приводы и соответствующие машиночитаемые носители обеспечивают энергонезависимые запоминающие устройства персонального
15 компьютера 20. Хотя описание вышеупомянутых машиночитаемых носителей относится к жесткому диску, сменному магнитному диску и диску CD-ROM, специалистам в данной области техники очевидно, что в иллюстративной операционной среде можно использовать и другие типы машиночитаемых носителей, например магнитные кассеты, карты флэш-памяти, цифровые видеодиски, картриджи Бернулли и т.п.

20 На дисках и в ОЗУ 25 могут храниться различные программные модули, в том числе операционная система 35, одна или несколько прикладных программ 200, 220, данные программ, например список 225 доступных пространств имен, и другие программные модули (не показаны).

Пользователь может вводить команды и информацию в персональный компьютер 20
25 через клавиатуру 40 и указательное устройство, например мышь 42. Другие устройства ввода (не показаны) могут включать в себя микрофон, джойстик, игровую панель, спутниковую антенну, сканер и т.п. Для подключения этих и других устройств ввода к процессору 21 обычно используется интерфейс 46 последовательного порта, подключенный к системной шине, но могут использоваться и другие интерфейсы, например
30 игровой порт или универсальная последовательная шина (USB). Монитор 47 или устройство отображения другого типа также подключен к системной шине 23 через интерфейс, например видеоадаптер 48. Помимо монитора, персональные компьютеры обычно включают в себя другие периферийные устройства вывода (не показаны), например громкоговорители или принтеры.

35 Персональный компьютер 20 может работать в сетевой среде, используя логические соединения с одним или несколькими удаленными компьютерами, например удаленным компьютером 49. Удаленный компьютер 49 может представлять собой сервер, маршрутизатор, одноранговое устройство или другое общее сетевое устройство и обычно включает в себя многие или все элементы, описанные применительно к персональному
40 компьютеру 20, хотя на фиг.1 изображено только запоминающее устройство 50. На сервере 49 и в памяти 50 могут размещаться прикладные программы, например приложение 220, и хранилище данных, например библиотека 230 пространств имен. Логические соединения, указанные на фиг.1, включают в себя локальную сеть (ЛС) 51 и глобальную сеть (ГС) 52. Такие сетевые среды обычно имеют место в учреждениях, компьютерных сетях
45 предприятия, интрасетях и Интернете.

При использовании сетевой среды ЛС персональный компьютер 20 подключен к ЛС 51
через сетевой интерфейс 53. При использовании в сетевой среде ГС персональный компьютер 20 обычно содержит модем 54 или иное средство установления связи через ГС 52, например, Интернету. Модем 54, который может быть внутренним или внешним,
50 подключен к системной шине 23 через интерфейс 46 последовательного порта. В сетевой среде программные модули, показанные применительно к персональному компьютеру 20, или некоторые из них могут храниться в удаленном запоминающем устройстве. Очевидно, что показанные сетевые соединения носят иллюстративный характер и что можно

использовать другие средства установления линий связи между компьютерами.

Работа

На фиг.2 изображена упрощенная блок-схема, иллюстрирующая взаимодействие между приложением-поставщиком и приложением-потребителем, при котором данные вырезают или копируют с использованием приложения-поставщика и вставляют с использованием приложения-потребителя. Согласно фиг.2 данные, вырезанные или скопированные из приложения-поставщика 200, сохраняются для использования приложением-потребителем 220. Приложение-поставщик может представлять собой любую прикладную программу, из которой данные могут быть вырезаны или скопированы для последующей вставки, в том числе приложение текстового редактора, приложение электронной таблицы, приложение слайдовой презентации и т.д. Когда пользователь выбирает данные с использованием приложения-поставщика 200, выполняется операция 205 вырезания или копирования, которая может включать в себя вырезание выбранных данных из документа-поставщика или может включать в себя копирование выбранных данных из документа-поставщика. Выбранные данные, будучи вырезаны или скопированы, переносятся в память 210, например буфер обмена 210, показанный на фиг.2. Вырезанные или скопированные данные хранятся в памяти 210 для последующей вставки исходным приложением-поставщиком 200 или приложением-потребителем 220 по выбору пользователя.

Согласно варианту осуществления настоящего изобретения буфер обмена 210 является примером ресурса памяти, поддерживаемого операционной системой с многооконным интерфейсом. В буфере обмена хранится копия скопированных или вырезанных данных, и последующая операция вставки переносит данные из буфера обмена 210 в программу-потребитель. Как известно специалистам в данной области техники, буфер обмена 210 позволяет переносить данные из одного приложения в другое, как описано выше, при условии, что второе приложение или приложение-потребитель 220 может читать данные, сгенерированные приложением-поставщиком 200. Чтобы гарантировать, что приложение-потребитель может читать, понимать и использовать данные, вырезанные или скопированные приложением-поставщиком 200, данные, вырезанные или скопированные приложением-поставщиком, обычно сохраняют в буфере обмена 210 в базовом формате, например в формате языка гипертекстовой разметки, который с большой вероятностью позволяет использовать их разнообразными приложениями-потребителями 220. Данные также можно сохранять в формате RTF или в неформатированном виде. Когда первое приложение осуществляет копирование, все эти форматы объявляются в буфере обмена. Затем приложение, осуществляющее вставку, должно решить, какой из доступных форматов желательно запросить.

Когда приложение-потребитель 220 запущено для вставления вырезанных или скопированных данных из приложения-поставщика 200, выбирают операцию вставки 215, чтобы вставить вырезанные или скопированные данные в нужное место документа, обрабатываемого приложением-потребителем. Следует понимать, что операцию вырезания/копирования 205 и операцию вставки 215 можно комбинировать в виде операции перетаскивания и сброса, когда выбранные данные перетаскивают из документа приложения-поставщика и сбрасывают в документ приложения-потребителя. Если вырезанные или скопированные данные были форматированы приложением-поставщиком согласно нескольким базовым типам форматирования, описанным выше, при желании, можно вставить вырезанные или скопированные данные в приложение-потребитель, но многие особенности, обеспеченные данным приложением-поставщиком, включая специальное форматирование, часто утрачиваются. Например, если пользователь вырезает или копирует данные, введенные в табличной форме в приложение-поставщик 200 текстового редактора, а затем вставляет эти данные в приложение-потребитель 220 электронной таблицы, данные, первоначально форматированные в табличном виде, могут быть просто вставлены в одну ячейку электронной таблицы, что не позволяет приложению электронной таблицы совершать над данными операции, необходимые пользователю приложения-потребителя 220 электронной таблицы.

С появлением языков разметки, например расширяемого языка разметки (XML), документы, составляемые в приложении-поставщике 200, нередко аннотируют с помощью структуры языка разметки, чтобы придать документу полезную структуру для управления, манипулирования данными, содержащимися в документе, и их представления. Как
 5 известно специалистам в данной области техники, разные прикладные программы содержат разное «собственное» программирование для чтения, понимания и использования языков разметки, например XML. Это значит, что приложение-поставщик 200 может быть запрограммировано на обеспечение других структур XML и функций или
 10 большего их количества по сравнению с приложением-потребителем 220. Соответственно, если данные вырезаны или скопированы из документа в формате XML, созданного приложением-поставщиком 200, для потребления отличным от него приложением-потребителем 220, то большая часть структуры и функций, связанных с форматированием XML вырезанных или скопированных данных, может быть утрачена, когда приложение-потребитель 200 вставляет данные в документ, созданный и обрабатываемый
 15 приложением-потребителем 220.

Ниже приведен пример структуры XML, которую можно применить к статье, написанной пользователем в приложении-поставщике 200 текстового редактора.

Образец структуры XML

20

```

    <article>
      <title>EXTRA EXTRA</title>
      <body>
        <summary>Today in New York...</summary>
        <main> Today in New York...</main>
      </body>
      *
      *
      *
    </article>
  
```

25

Как показано в образце структуры XML, ряд элементов XML аннотируют документ «статья». Например, в начало и конец документа включен тег (неотображаемый элемент разметки документа) <article>, определяющий документ как документ «статья», а внутри структуры «статья» включены различные другие элементы, например элемент <title> и элемент <body>. Внутри элемента <body> имеются два дочерних
 30 элемента, <summary> и <main>, элемента <body>. XML-аннотация документа позволяет пользователю задавать участки документа для определенных типов данных и структуры данных. Например, пользователь может задать участок документа внутри
 35 элемента <title>, чтобы включить данные типа «название» и включить предписанную допустимую структуру для названия.

Чтобы снабдить документ набором правил для грамматики и типов данных, регламентирующих типы и структуру данных, которые могут содержаться в данном документе, например, проиллюстрированном выше документе «статья», к документу присоединяют или с ним связывают схему XML, обеспечивающую правила, регламентирующие каждый из элементов XML и тегов, с помощью которых пользователь может аннотировать данный документ. Например, документ «статья» может иметь
 40 присоединенную или связанную с ним схему, например article-schema.xsd, обеспечивающую допустимый набор элементов XML, состоящий, например, из элемента <article>, элемента <title>, элемента <body> и т.д. Схема содержит правила, определяющие порядок применения этих элементов к документу, и конкретные правила, связанные с отдельными элементами, применяемыми к документу. Например, схема,
 45 присоединенная или связанная с документом «статья», может предписывать, что данные, связанные с данным элементом, например элементом <date> (в вышеприведенном примере не показан), должны содержать элемент «число», затем элемент «месяц», а затем элемент «год». Дополнительно, схема может требовать, чтобы элемент <date> следовал

50

сразу после элемента <title>.

Специалистам в данной области техники известно, что разработчики схем XML определяют имена элементов XML и соответствующие типы данных и структуры данных, допустимые для этих элементов. Соответственно, все пользователи документов, аннотированных структурой XML согласно данной схеме, могут использовать данные, содержащиеся в структуре XML, независимо от типа и структуры документа в целом. Например, если проиллюстрированный выше документ «статья» поступает издателю документа, то издатель может разработать прикладные программы для синтаксического анализа документа, позволяющего определить местоположение конкретных типов данных в документе, которые издатель может использовать. Например, издатель может пожелать опубликовать только название статьи в качестве анонса будущей публикации всей статьи. Идентификация XML осуществляется посредством пространства имен. Пространство имен обеспечивает идентификатор (ИД, ID), сообщающий любому потребителю XML тип потребляемого XML. Просматривая пространство имен, приложение-потребитель узнает, какая схема использовалась для создания файла XML. С использованием пространства имен, заданного в документе, издатель узнает, что данные, связанные с элементом XML <title>, были подготовлены согласно схеме, управляющей документом. Соответственно, издатель может разработать прикладную программу или файл преобразования XSLT для определения местоположения элемента <title> и для извлечения связанных с ним данных для вставки в собственный документ издателя (в том числе, в каком-то особом формате), чтобы использовать данные в качестве анонса будущей публикации всей статьи.

Следуя этому примеру, несколько разных издателей могут подписаться на то же пространство имен, определяющее правила, связанные с документом «статья», что позволит каждому издателю затем получить документ «статья» от автора документа и использовать данные, содержащиеся в статье, согласно элементам XML, структурирующим данные. Таким образом, первая издательская компания может вставить только данные, содержащиеся в элементе <summary>, а второй издатель может быть заинтересован в извлечении только данных, содержащихся в элементе <title>. Каждый издатель может извлекать нужные ему данные вне зависимости от других аспектов или элементов документа, используя свои собственные прикладные программы или файлы преобразования XSLT для определения местоположения нужных данных согласно структуре XML. Это возможно благодаря тому, что каждый пользователь документа следует правилам относительно типа данных и структуры данных, предписанным в пространстве имен, присоединенном к документу или связанном с ним.

Если приложение-потребитель находит файл преобразования XSLT для использования с одним из доступных пространств имен, то выбранные данные преобразуются файлом преобразования XSLT, и приложение-потребитель принимает результаты как часть функции вставки. Также может быть случай, когда приложение-потребитель знает файл преобразования и потребляет доступное пространство имен, самостоятельно осуществляя преобразование этих данных. Например, согласно описанному выше, если приложение-поставщик является приложением электронной таблицы и выбранные данные форматированы в две строки и три столбца, то файл преобразования XSLT, преобразующий данные электронной таблицы для потребления, например, приложением текстового редактора, допускает такое преобразование данных, что приложение текстового редактора может вставлять данные в виде двух списков из трех элементов каждый, а не в виде необработанных данных без форматирования или в табличной форме, используемой в электронной таблице. Заметим, что документ первого приложения или приложения-поставщика можно размечать согласно индивидуальной схеме XML. Например, данные можно размечать согласно схеме «курс акций». В этом случае, если выбранные данные скопированы для вставки, например, из приложения электронной таблицы в приложение текстового редактора, то при записи выбранных данных в память с целью вставки в приложение текстового редактора с выбранными данными будут связаны

пространство имен, связанное со схемой «курс акций», а также пространство имен, связанное видимым расположением электронной таблицы.

Распознав схему «курс акций», приложение-потребитель обращается к списку доступных ресурсов и обнаруживает, что каждая ячейка выбранных данных является символом фондов компании и что для каждой компании имеется соответствующее описание компании. Соответственно, приложение-потребитель будет определять местоположение ресурса, например файла преобразования XSLT, в списке доступных пространств имен или библиотеке пространств имен, который указывает приложению-потребителю, как вставлять размеченные данные. Пользовательский интерфейс, предоставленный пользователю приложения-потребителя, может обеспечивать опцию «вставить краткую характеристику компании». Когда пользователь выбирает «вставить краткую характеристику компании», файл преобразования XSLT, связанный с этой функцией вставки, может позволить приложению-потребителю вставлять символы фондов компании, выбранные из приложения-поставщика, совместно с краткой характеристикой компании, которую можно импортировать через функцию вставки, выбранную пользователем из доступных пространств имен, связанных с выбранными данными. Это решение лучше, чем альтернатива выбора пространства имен электронной таблицы и вставки данных в том же виде, в каком они были представлены в электронной таблице.

Те, кто знаком с расширяемым языком разметки, знает, что пространства имен XML обеспечивают способ квалифицирования элементов и имен атрибутов, используемых в документах XML, путем сопоставления этих элементов и имен атрибутов с пространствами имен, идентифицируемыми ссылками универсального индикатора ресурса (URI). Пространства имен XML представляют собой совокупности имен, идентифицируемые ссылками URI, которые используются в документах XML как типы элементов и имена атрибутов. Единичный документ XML может содержать элементы и атрибуты, определенные для множественных программных модулей и используемые ими. Например, согласно варианту осуществления настоящего изобретения единичный элемент XML, например документ текстового редактора, может содержать элементы и атрибуты, определенные для разных программных модулей и используемые ими. Например, документ текстового редактора может иметь элементы и атрибуты, определенные для модуля обработки HTML, модуля обработки XML приложения текстового редактора и используемые ими, или документ может содержать элементы и атрибуты, определенные для одного или нескольких файлов схемы, связанных с документом, и используемые ими или связанные с ними. Например, элементы и атрибуты могут быть связаны с документом текстового редактора, чтобы документ был связан с файлом схемы, связанным с документом резюме, юридическим документом и т.п. Соответственно, отдельный документ, например иллюстративный документ текстового редактора, может иметь пространство имен, идентифицирующее типы элементов и имена атрибутов, связанные с каждым из различных программных модулей, которые могут потреблять или использовать данные из документа. Как и в вышеприведенных примерах, документ текстового редактора может содержать пространство имен, связанное с модулем обработки HTML, пространство имен, связанное с модулем обработки XML текстового редактора, и пространство имен, связанное с файлами схемы документа резюме или юридического документа.

Согласно варианту осуществления настоящего изобретения при выборе данных для вставки в документ второго приложения, информация, идентифицирующая пространства имен, связанные с выбранными данными, обеспечивается совместно с выбранными данными, чтобы информировать второе приложение или приложение-потребитель о пространствах имен, связанных с выбранными данными, чтобы второе приложение или приложение-потребитель могло оперировать типами элементов и/или атрибутами, связанными с выбранными данными, согласно программированию второго приложения. Например, если первое приложение является приложением текстового редактора, а второе приложение является приложением электронной таблицы, то приложение электронной таблицы может использовать информацию, идентифицированную пространствами имен,

связанными с выбранными данными, чтобы определить, что выбранные данные будут оптимально вставлены в документ второго приложения с использованием файла преобразования XSLT, связанного с типами элементов и атрибутами, связанными с одним из заданных пространств имен. Например, если одно из данных пространств имен

5 идентифицирует типы элементов и атрибуты, связанные с манипулированием данными, содержащимися в выбранных данных, а другие пространства имен связаны с отображением выбранных данных, то второе приложение, в данном примере приложение электронной таблицы, может выбрать пространство имен, связанное с манипулированием

10 данными, а не пространство имен, связанное с отображением

После того как приложение-потребитель обнаружит пространства имен, связанные с данными XML, выбранными из документа первого приложения, согласно варианту осуществления настоящего изобретения приложение-потребитель может обратиться к

15 списку доступных пространств имен или к библиотеке пространств имен, чтобы найти ресурсы, которые приложение-потребитель может использовать для преобразования выбранных данных для оптимального потребления вторым приложением или

приложением-потребителем. Например, если второе приложение является приложением электронной таблицы, которое пытается вставить данные, выбранные из приложения

20 текстового редактора, то второе приложение может просматривать список доступных пространств имен или библиотеку пространств имен, чтобы получить файл преобразования XSLT, который второе приложение может использовать для преобразования данных XML

текстового редактора в данные XML электронной таблицы для оптимального потребления приложением электронной таблицы. Специалистам в данной области техники понятно, что расширяемый язык стилей (XSL) содержит словарь XML для задания форматирования

25 данных. XSL задает стилистику на документе XML с использованием файлов информации XML, чтобы описывать, как документ преобразуется из одного документа XML в другой документ XML или документ другого типа, например документ HTML.

Согласно фиг.2 пространства имен можно присоединять к документу, подготовленному приложением-поставщиком, или пространство имен можно поддерживать в другом месте,

30 например библиотеке 230 пространств имен, доступной для документа. Документ может содержать указатель пути к файлу или уникальный идентификатор пространства имен (например, универсальный идентификатор ресурса или универсальное имя ресурса) для указания местоположения и/или идентификации пространства имен. Подробное описание библиотеки 230 пространства имен можно найти в патентной заявке США, озаглавленной

35 «Система и способ для обеспечения информации относительно пространства имен» ("System and Method for Providing Namespace Related Information"), за № 10/184,190, поданной 27 июня 2002 г. и включенной в настоящее описание посредством ссылки во всей полноте. Подробное описание способа загрузки пространства имен из библиотеки 230

пространств имен для использования приложением-поставщиком 200 или приложением-потребителем 200 можно найти в патентной заявке США, озаглавленной «Механизм

40 загрузки компонентов программного обеспечения из удаленного источника для использования локальной прикладной программой» ("Mechanism for Downloading Software Components from a Remote Source for Use by a local Software Application"), за № 10/164,260, поданной 5 июня 2002 г. и включенной в настоящее описание посредством

45 ссылки во всей полноте.

Когда пользователь загружает или иным образом получает пространство имен и связанную с ним информацию для использования с документами, созданными

пользователем, пространство имен можно загрузить в список доступных пространств имен, состоящий из этого пространства имен и его ресурсов, а также любого другого

50 пространства имен, которое уже было установлено (инсталлировано). Список 225 поддерживается на компьютере 20 пользователя или доступен пользователю в удаленном месте хранения, например на сервере 49, работающем в распределенной вычислительной среде. Список доступных пространств имен может содержать пространства имен,

связанные с данными, содержащимися в документе приложения-поставщика, и пространства имен могут содержать информацию, позволяющую приложению-потребителю определять местоположение ресурсов и получать ресурсы, например преобразование XSLT, помогающее приложению-потребителю преобразовывать выбранные данные для

5 потребления приложением-потребителем.

Пусть, например, пользователь вырезает или копирует данные, подготовленные в табличной форме с использованием приложения-поставщика 200 текстового редактора. До вырезания или копирования данных данные структурируются с помощью аннотаций XML согласно собственному программированию XML приложения-поставщика 200. Если

10 приложение-потребитель 220 посредством операции 215 вставки должно вставить вырезанные или скопированные данные из буфера обмена 210, то приложение-потребитель 220 может вызвать список 225 доступных пространств имен, чтобы увидеть, имеется ли пространство имен или надлежащий файл преобразования XSLT для одного из

15 пространств имен для расширения собственного программирования XML приложения-потребителя 220, чтобы приложение-потребитель 220 могло вставить данные, вырезанные или скопированные из приложения-поставщика 200 текстового процессора в том же формате (например, в табличном формате), который был применен к этим данным с использованием приложения-поставщика 200. Например, в списке 225 доступных пространств имен может быть найден файл ресурса (в данном случае, преобразование XSLT) например, "wordprocessor-to-spreadsheet-schema.xsl", который приложение-

20 потребитель может использовать для преобразования из доступного пространства имен текстового редактора в пространство имен, понятное электронной таблице. Соответственно, когда приложение-потребитель получает дополнительное преобразование XSLT, собственное программирование XML приложения-потребителя расширяется, что

25 позволяет приложению-потребителю (например, приложению электронной таблицы) использовать данные, полученные от приложения-поставщика, согласно преобразованию пространства имен, полученному из списка доступных пространств имен. Таким образом, благодаря получению пространства имен XML из списка доступных пространств имен, приложение-потребитель принимает правила для грамматики XML и данных, необходимые

30 для обработки данных, полученных из приложения-поставщика, таким же образом, как эти данные создавало и обрабатывало приложение-поставщик.

Согласно вариантам осуществления настоящего изобретения, если данные структурированы с использованием приложения-поставщика согласно заданному пространству имен XML, то пользователь посредством приложения-потребителя может

35 вставить вырезанные или скопированные данные в приложение-потребитель согласно выбранному файлу преобразования XSLT третьей стороны. Например, если пользователь приложения-потребителя 220 принимает данные, вырезанные из вышеописанного документа «статья», то пользователь приложения-потребителя 220 может вызвать ресурсы списка доступных пространств имен, чтобы определить, имеются ли файлы

40 преобразования, связанные с разными издателями, позволяющие структурировать документ в приложении-потребителе, согласно требованиям конкретного издателя. Пользователь приложения-потребителя может затем вызвать ресурсы списка 225 доступных пространств имен для файла преобразования, которые позволят приложению-потребителю 220 использовать данные согласно нужному пространству имен. Если нужное

45 пространство имен отсутствует в списке 225 доступных пространств имен, пользователь может вызвать локальную или удаленную библиотеку 230 пространств имен, чтобы определить местоположение дополнительных ресурсов для пространств имен, которые приложение-потребитель может использовать, чтобы оперировать данными, полученными из приложения-поставщика 200.

50 На фиг.3 изображена упрощенная блок-схема пользовательских интерфейсов для предоставления пользователям расширенных функциональных возможностей вставки согласно иллюстративному варианту осуществления настоящего изобретения. Когда пользователь приложения-потребителя 220 выбирает операцию 215 вставки приложения-

потребителя 220, пользователю могут быть предоставлены различные опции вставки. Например, пользователь может выбрать операцию простой вставки, которая вставляет данные, вырезанные или скопированные из приложения-поставщика, в документ, используемый в приложении-потребителе, согласно форматированию, принятому по

5 умолчанию, например форматированию HTML. Если пользователь желает воспользоваться расширенными функциями вставки, отвечающими вариантам осуществления настоящего изобретения, то пользователь может выбрать альтернативную функцию вставки, например функцию специальной вставки, показанную на фиг.3.

Пользовательский интерфейс 310 специальной вставки иллюстрирует список опций
10 вставки, доступных пользователю. Например, пользователь может выбрать вставку данных в формате RTF или пользователь может выбрать вставку документа согласно формату HTML 320. Как показано в пользовательском интерфейсе 310, пользователь может также выбрать вставку данных согласно формату 325 XML без расширенных функций настоящего изобретения. Это значит, что приложение-потребитель 220 вставляет вырезанные или
15 скопированные данные согласно программированию XML, которое является собственным для приложения-потребителя 220. Если пользователь желает вставить данные согласно некоторой расширенной функции, соответствующей вышеописанному, то пользователь может выбрать кнопку 330 «другая» пользовательского интерфейса 310, чтобы запустить пользовательский интерфейс 350 доступных схем. С помощью пользовательского
20 интерфейса 350 доступных схем пользователь может выбрать доступный список 355, чтобы проверить, какие пространства имен и соответствующие файлы XSLT имеются в списке 225 доступных пространств имен, согласно описанному выше со ссылкой на фиг.2. Соответственно, выбор данного файла преобразования обеспечивает приложение-потребитель, используемое пользователем, расширенными функциями выбранного файла
25 XSLT, позволяющими оперировать данными, вырезанными или скопированными из приложения-поставщика, как если бы эти данные использовались или форматировались приложением-поставщиком.

Если список доступных пространств имен не содержит описанных пространств имен или
иных ресурсов, то приложение может также запустить функцию 370 библиотеки
30 пространств имен для обеспечения более обширного списка доступных визуальных представлений данных, которые можно получить из локальной или удаленной библиотеки 230, описанной выше со ссылкой на фиг.2. Если пользователь выбирает доступное визуальное представление из библиотеки 230 пространств имен через пользовательский интерфейс 370, то пользователю представляется пользовательский интерфейс 310
35 специальной вставки с обновленным списком доступных функций вставки, включающим в себя функцию вставки, разрешенную выбранным пространством имен. Следует понимать, что визуальное представление в одном случае может являться произвольной комбинацией пространств имен и связанных с ними преобразований XSLT. Преобразование XSLT будет преобразовывать доступное пространство имен в пространство имен, которое приложение-
40 потребитель, по своей сущности, понимает. Другие пространства имен обычно представляют другие типы данных, и другие преобразования XSLT представляют другие визуальные представления на этих других типах данных. Согласно фиг.3 при выборе XSLT 380 «Издатель А» в пользовательском интерфейсе 370 библиотеки пространств имен пользовательский интерфейс 310 специальной вставки заполняется функцией 390 вставки
45 «Издатель А». Соответственно, пользователь может выбрать функцию 390 вставки «Издатель А», чтобы применить загруженное преобразование 380 «Издатель А», позволяющее приложению-потребителю 220 вставить данные, полученные от приложения-поставщика 200, согласно файлу XSLT, обеспеченному издателем А. Преобразование идентифицируется как связанное с преобразованием пространства имен «Издатель А» в
50 пространство имен, понимаемое приложением-потребителем. Специалистам в данной области техники должно быть понятно, что конкретные файлы XSLT, описанные здесь со ссылкой на фиг.2 и 3, носят исключительно иллюстративный характер, поскольку любые другие файлы XSLT или любой другой способ преобразования или интерпретации

конкретного пространства имен можно сделать доступным приложению-потребителю 220 для обеспечения расширенных функций вставки вырезанных или скопированных данных, полученных из приложения-поставщика 200.

На фиг.4 показано отображение на экране компьютера программного приложения, вставляющего и потребляющего данные, вырезанные или скопированные из приложения-поставщика. Снимок 400 экрана показывает иллюстративное приложение текстового редактора, в котором пользователь готовит документ 410. Согласно варианту осуществления настоящего изобретения пользователь вставил данные в таблицу, которая была скопирована из приложения-поставщика 200 электронной таблицы. При выполнении операции вставки для вставки данных в документ 410 пользователю предоставляется диалоговое окно 420, показывающее пользователю три разные формата данных, включая формат 425 «История за восемь месяцев», формат 430 «Запланированная финансовая сводка» и формат 435 «Анализ промышленности». Заметим, что диалоговое окно 420 является альтернативным диалоговым окном к диалоговому окну 310, показанному на фиг.3. Согласно иллюстративным данным, показанным на фиг.4, пользователь выбрал формат 425 «История за восемь месяцев», который является визуальным отображением одного из пространств имен в списке 225 доступных пространств имен, чтобы форматировать данные, полученные из приложения-поставщика (например, приложения электронной таблицы), в документе 410 согласно файлу XSLT, задающему структуру XML для «Истории за восемь месяцев». Следует понимать, что в списке доступных пространств имен или библиотеке 225 пространств имен может быть доступен ресурс преобразования, например "eightmonthhistory-schema.xsl", позволяющий приложению-потребителю поддерживать структуру XML для обеспечения форматирования «Таблица 1», показанного на фиг.4. Если бы пользователь выбрал вырезание и копирование информации из приложения-поставщика, относящейся к формату «Анализ промышленности», то пользователь мог бы выбрать формат 435 «Анализ промышленности», чтобы данные, вырезанные или скопированные из приложения-поставщика, форматировались в приложении-потребителе согласно формату, в котором данные были структурированы приложением-поставщиком.

Согласно варианту осуществления настоящего изобретения, когда данные вырезают или копируют в формате XML и сохраняют в буфере обмена 210 для последующей вставки приложением-потребителем 220, можно использовать три или более форматов, чтобы помочь приложению-потребителю определить, как нужно обрабатывать вырезанные или скопированные данные согласно функциям настоящего изобретения. Три формата включают в себя CF_XML_HEADER, CF_XML_VIEW и CF_XML_DATA. Согласно описанному выше многие приложения, включая приложение-поставщик 200 и приложение-потребитель 220, поддерживают структуру разметки XML и файлы, созданные этими приложениями. Однако, как описано выше, поддержка XML одного приложения часто бывает несовместима с поддержкой XML другого приложения, вследствие чего форматирование и другие особенности, связанные с XML, теряются при копировании данных из одного приложения в файлы приложения-потребителя. Дополнительно, как описано выше, существуют также схемы XML, которые обеспечиваются третьими сторонами для использования различными приложениями, но не поддерживаются природным программированием XML приложения.

Формат CF_XML_HEADER («ЗАГОЛОВОК») содержит информацию о том, какое пространство имен доступно копирующему приложению. Все вставляющие приложения могут обращаться к CF_XML_HEADER, чтобы решить, имеется ли одно или несколько пространств имен, которые они способны потреблять. ЗАГОЛОВОК также идентифицирует, где можно найти разные пространства имен, чтобы вставляющее приложение, решив, какое пространство имен или какие пространства имен оно хочет потреблять, могло выполнить обратный вызов, чтобы получить нужные данные. На месте ЗАГОЛОВОК может идентифицировать формат CF_XML_DATA как содержащий нужное(ые) пространство(а) имен. Информация является полезной для приложения-потребителя, поскольку дает

приложению-потребителю возможность определять, какие форматы желательно потреблять. Это может быть полезно в отношении производительности, поскольку копирующему приложению не требуется обеспечивать все данные в разных форматах, пока эти конкретные форматы не запрошены. В остальной части этого описания, будем полагать, что доступно два формата.

Первый формат буфера обмена - это CF_XML_VIEW, имеющий одно или несколько пространств имен и обычно имеющий скорее разметки форматирования, чем разметки данных (например, "spreadsheetML; или documentML"). Второй формат буфера обмена - это CF_XML_DATA, который содержит одно или несколько пространств имен, которые не задают форматирование или вид, но задают смысловое содержание данных. Эти два формата в действии можно легко видеть с использованием примера, в котором используется программа электронной таблицы. Если эта программа электронной таблицы поддерживает структуры XML, то можно представить себе, что пространство имен, задающее финансовые данные, можно применять к этой электронной таблице, чтобы идентифицировать разные ячейки и их отношение к финансовым данным. Когда приложение электронной таблицы осуществляет операцию копирования, оно может представлять данные двумя способами. Первый способ состоит в том, чтобы представлять их по мере появления в электронной таблице и в использовании языка разметки электронной таблицы. Язык разметки электронной таблицы задает общие признаки электронной таблицы, например форматирование, размещение и т.д. Эти данные можно размещать в буфере обмена в формате CF_XML_VIEW, поскольку он наилучшим образом задает, как «выглядят» данные. Это может быть полезно для человека, вставляющего данные, если ему важнее внешний вид и размещение, чем смысловая нагрузка данных.

Второй способ представления данных в буфере обмена состоит в использовании пространства имен финансовых данных. Это пространство имен не имеет информации, касающейся вида или размещения данных, но зато описывает, что означают сами данные. Эти данные размещаются в буфере обмена в формате CF_XML_DATA. В этом случае приложению-потребителю приходится самостоятельно решать, как лучше отображать эти данные, поскольку информация расположения недоступна. Например, в случае вставки в приложение текстового редактора это может быть более желательный формат. Дело в том, что вид финансовых данных в электронной таблице может полностью отличаться от того, как они должны выглядеть в приложении текстового редактора. Приложению текстового редактора придется обращаться к пространству имен финансовых данных, а затем находить некоторую информацию вида, например преобразование XSLT, помогающее ему правильно отображать данные.

Разделение разных пространств имен на разные форматы буфера обмена, значительно повышает эффективность, поскольку приложение-потребитель может не нуждаться в одном из или обоих форматах «ВИД» и «ДАННЫЕ» из приложения-поставщика. Таким образом, если на основании функции вставки, выбранной пользователем, приложение-потребитель определяет, что ему не нужна информация, содержащаяся в каком-либо из форматов «ВИД» или «ДАННЫЕ», то приложение-потребитель не будет загружать данные из этих форматов для использования приложением-потребителем. Если приложение-потребитель является приложением того же типа (например, приложением текстового редактора), что и приложение-поставщик, то приложению-потребителю могут не потребоваться дополнительные функциональные возможности вставки для осуществления операции вставки, выбранной пользователем. Формат CF_XML_HEADER содержит информацию о версии и пространстве имен для обоих форматов «ДАННЫЕ» и «ВИД». Формат заголовка также содержит информацию о размере каждого из файлов формата ДАННЫЕ и ВИД. Ниже приведен пример формата CF_XML_HEADER, связанного с фрагментом данных, скопированных из файла текстового редактора, размеченного с помощью схемы XML "Get A Job.com" третьей стороны.

```
<cfh:header cfh:Version="1.0" xmlns:cfh="urn:schemas-microsoft-com:office:clipboard:header">
```

5

```

<cfh:data cfh:bytes="00000001343">
<cfh:Item cfh:name="D1" cfh:bytes="00000001343">
  <cfh:URI cfh:value="urn:schemas-getajob-
    com:resume" cfh:root="yes"/>
    </cfh:Item>
  </cfh:data>
  <cfh:view cfh:bytes="00000003612">
<cfh:Item cfh:name="V1" cfh:bytes="00000003612">
  <cfh:URI cfh:value="urn:schemas-microsoft-
    com:office:word.1.0" cfh:root="yes"/>
  <cfh:URI cfh:value="urn:schemas-microsoft-
    com:schemaLibrary"/>
  <cfh:URI cfh:value="urn:schemas-microsoft-
    com:AML"/>
  </cfh:Item>
</cfh:view>
</cfh:header>

```

10

15

20

25

Формат данных CF_XML_VIEW («ВИД») предоставляет информации приложению-потребителю в качестве собственного программирования XML приложения-поставщика. Считывая данные, содержащиеся в формате «ВИД», приложение-потребитель может определить, способно ли оно согласно своему собственному программированию XML полностью использовать собственное программирование XML приложения-поставщика без необходимости получать дополнительные файлы преобразования XSLT при выполнении операции вставки. Ниже приведен пример формата заголовка CF_XML_VIEW, который может быть связан с данными, вырезанными или скопированными из приложения-поставщика.

30

```

<cb:view xmlns:cb="urn:schemas-microsoft-
com:office:clipboard:view"> <!--Корневой элемент для буфера
  обмена-->
  <cb:item cb:name="V0" cb:URI="URI корневой схемы">

```

35

Примечание: Согласно одному варианту осуществления здесь могут быть данные. Поскольку эти данные располагаются перед тегом cb:start, приложение-потребитель знает, что эти данные не являются частью скопированного фрагмента, но все же могут обеспечить представляющую интерес информацию о сущности источника в целом.

<cb:start/> <!--Единичный тег, подлежащий размещению в строке, указывающей начало фрагмента-->

40

Примечание: Согласно одному варианту осуществления здесь могут быть данные. Поскольку они располагаются после тега cb:start, приложение-потребитель знает, что эти данные являются частью скопированного фрагмента.

<cb:end/> <!--Единичный тег, подлежащий размещению в строке, указывающей конец фрагмента-->

45

Примечание: Согласно одному варианту осуществления здесь могут быть данные. Поскольку эти данные располагаются после тега cb:end, приложение-потребитель знает, что эти данные не являются частью скопированного фрагмента, но все же могут обеспечить интересующую информацию о природе источника в целом.

</cb:item>

<!--Возможно поместить более одного, если вставляющее приложение решит: -->
 <cb:item cb:name="V1" cb:URI="URI корневой схемы">

50

```

<cb:start/> <!--Единичный тег, подлежащий размещению
  в строке, указывающей начало фрагмента-->
<cb:end/> <!--Единичный тег, подлежащий размещению
  в строке, указывающей конец фрагмента-->
  </cb:item>

```

Формат CF_XML_DATA («ДАННЫЕ») предоставляет информацию приложению-потребителю о пространствах имен XML, связанных с данными, вырезанными или скопированными из приложения-поставщика, где эти пространства имен XML не являются частью собственного программирования XML приложения-поставщика. Например, заголовок «ДАННЫЕ» может предоставлять копирующему приложению информацию, касающуюся пространств имен XML, связанных со схемой, созданной пользователем приложения-поставщика, или информацию, касающуюся пространства имен XML, связанного со схемой третьей стороны, например издателя, или схемой брокерской компании, используемую при структурировании данных для потребления или представления пользователями, как описано выше. Ниже приведен пример формата CF_XML_DATA, который можно передавать приложению-потребителю, чтобы снабжать его информацией, касающейся несобственных пространств имен XML, связанных с данными, вырезанными или скопированными приложением-поставщиком.

```
<cb:data xmlns:cb="urn:schemas-microsoft-com:office:clipboard:data"> <!-- Корневой элемент для буфера обмена-->
```

```
<cb:item cb:name="D0" cb:URI="URI корневой схемы">
```

Примечание: Согласно одному варианту осуществления здесь могут быть данные. Поскольку эти данные располагаются перед тегом cb:start, приложение-потребитель знает, что эти данные не являются частью скопированного фрагмента, но все же могут обеспечить интересующую информацию о природе источника в целом.

```
<cb:start/> <!--Единичный тег, подлежащий размещению в строке, указывающей начало фрагмента-->
```

Примечание: Согласно одному варианту осуществления здесь могут быть данные. Поскольку они располагаются после тега cb:start, приложение-потребитель знает, что эти данные являются частью скопированного фрагмента.

```
<cb:end/> <!--Единичный тег, подлежащий размещению в строке, указывающей конец фрагмента-->
```

Примечание: Согласно одному варианту осуществления здесь могут быть данные. Поскольку эти данные располагаются после тега cb:end, приложение-потребитель знает, что эти данные не являются частью скопированного фрагмента, но все же могут обеспечить интересующую информацию о природе источника в целом.

```
</cb:item>
```

```
<!--Возможно поместить более одного, если вставляющее приложение решит: -->
```

```
<cb:item cb:name="D1" cb:URI="URI корневой схемы">
```

```
<cb:start /> <!--Единичный тег, подлежащий размещению в строке, указывающей начало фрагмента-->
```

```
<cb:end/> <!--Единичный тег, подлежащий размещению в строке, указывающей конец фрагмента-->
```

```
</cb:item>
```

```
</cb:data>
```

На фиг.5 и 6 показаны логические блок-схемы, иллюстрирующие способ расширения функций вставки приложения-потребителя согласно варианту осуществления настоящего изобретения. Способ 500 начинается на начальном этапе 505 и переходит к этапу 510, на котором данные из приложения-поставщика вырезаются, копируются и сохраняются в буфере обмена 210 для последующей вставки приложением-поставщиком 220. На этапе 515 пользователь выбирает вставку из приложения-потребителя 220 для вставки вырезанных или скопированных данных в документ, используемый приложением-потребителем. На этапе 520 приложение-потребитель определяет, связан ли с вырезанными или скопированными данными формат CF_XML_HEADER. Если доступных форматов CF_XML_HEADER нет, то способ переходит к этапу 575, фиг.6, и вставка данных производится без какой-либо расширенной функции вставки. В соответствии с вышеизложенным заметим, что, если дополнительные схемы XML и/или структуры данных

доступны для приложения-потребителя, то эта информация предоставляется приложению-потребителю через формат CF_XML_HEADER.

Если на этапе 520 определено, что формат «ЗАГОЛОВОК» присутствует, то способ переходит к этапу 525 для определения того, содержат ли данные формат CF_XML_VIEW.

5 Если да, то способ переходит к этапу 530, и приложение-потребитель создает список доступных пространств имен, указанных в заголовке, которые можно использовать для вставки вырезанных или скопированных данных. Например, если приложение-поставщик является приложением электронной таблицы, то приложению-потребителю может потребоваться искать в библиотеке пространств имен доступный файл преобразования или
10 другой ресурс, помогающий приложению-потребителю (например, текстовому редактору) вставлять вырезанные или скопированные данные в соответствии с функциями XML приложения-поставщика. Причиной обращения к библиотеке пространств имен является то, что приложение-потребитель не понимает доступного пространства имен и должно найти дополнительную информацию о том, как обрабатывать их. Если пространства имен
15 XML отсутствуют в списке доступных пространств имен, то способ переходит к этапу 550 и приложение-потребитель ищет в библиотеке 230 пространств имен дополнительные ресурсы, которые могут помочь вставлять вырезанные или скопированные данные, полученные от приложения-поставщика.

Если на этапе 535 определено, что формат CF_XML_DATA обеспечен, то способ
20 переходит к этапу 540, на котором приложение-потребитель вызывает список доступных пространств имен, чтобы получить файлы преобразования и/или другие ресурсы, связанные с пространствами имен, созданными в приложении-поставщике или полученными от источника третьей стороны, внешнего по отношению к приложению-поставщику, как описано выше. Следует понимать, что вызов списка доступных
25 пространств имен, осуществляемый на этапах 530 и 540, может производиться путем предоставления пользовательского интерфейса, как описано выше со ссылкой на фиг.3.

На этапе 545, фиг.6, производится определение, понимает ли приложение-потребитель пространства имен и соответствующие файлы преобразования, указанные для данных, вырезанных или скопированных из приложения-поставщика. Например, если приложение-
30 потребитель является приложением текстового редактора и приложение-поставщик является приложением текстового редактора, то приложению-потребителю может не потребоваться дополнительных ресурсов в виде файлов преобразования для обеспечения форматирования XML, связанного с вырезанными или скопированными данными. В этом случае способ переходит к этапу 575, и приложение-потребитель вставляет вырезанные
35 или скопированные данные без расширения функций.

Если приложение-потребитель не понимает пространств имен и соответствующих ресурсов, связанных с выбранными данными, или если нет пространств имен или ресурсов, доступных для приложения-потребителя, то способ переходит к этапу 550, и пользователю предоставляется доступ к библиотеке 230 пространств имен для выбора дополнительных
40 пространств имен и ресурсов, связанных с данными, содержащимися в формате CF_XML_DATA, полученном от приложения-поставщика. Если формат «ДАННЫЕ», полученный от приложения-поставщика, указывает пространство имен третьей стороны, связанное с вырезанными или скопированными данными, например файл XSLT данного издателя, то приложение-потребитель может получить файл XSLT, необходимый
45 приложению-потребителю для вставки данных согласно указанному пространству имен, из библиотеки 230 пространств имен. На этапе 555 производится определение того, выбрал ли пользователь какие-либо доступные файлы преобразования для использования приложением-потребителем. Если нет, то способ переходит к этапу 560, и функции вставки пользовательского интерфейса 310 заполняются функциями, принятыми по
50 умолчанию, как показано на фиг.3.

При наличии ресурсов, например файлов преобразования XSLT, либо в списке доступных пространств имен, либо в библиотеке пространств имен способ переходит к этапу 565, и пользовательский интерфейс 310 вставки заполняется дополнительными

функциями, выбранными пользователями, как описано выше со ссылкой на фиг.3. На этапе 570 производится определение того, выбирает ли пользователь принятую по умолчанию базовую функцию вставки, или пользователь выбирает специальную функцию вставки. Если пользователю нужна специальная функция вставки, то способ переходит к этапу 580, и пользователь выбирает нужную специальную функцию вставки. Способ переходит к этапу 575, на котором вставка данных осуществляется согласно функции вставки, выбранной пользователем. Если на этапе 570 определено, что пользователь не желает выбирать специальную функцию вставки, то способ переходит к этапу 575, и вставка данных осуществляется без расширенных функций вставки, по выбору пользователя. Способ заканчивается на этапе 595.

Согласно вышеприведенному описанию обеспечены способы и системы для расширения функциональных возможностей вставки, доступных компьютерному программному приложению для вставки данных в документ, создаваемый на компьютере. Специалистам в данной области техники очевидно, что настоящее изобретение допускает различные модификации и вариации, не выходящие за рамки сущности и объема изобретения. Другие варианты осуществления изобретения будут очевидны специалистам в данной области техники из рассмотрения описания и практики применения раскрытого здесь изобретения.

Формула изобретения

1. Способ подготовки данных для вставки во второй документ, содержащий этапы, на которых открывают первый документ посредством приложения-поставщика, выбирают данные из первого документа для вставки во второй документ посредством приложения-потребителя, сохраняют выбранные данные в области памяти, выбирают совокупность пространств имен, доступных для приложения-поставщика, связывают эту совокупность пространств имен с выбранными данными из приложения-поставщика, сохраняют упомянутую совокупность пространств имен, связанных с выбранными данными, в упомянутой области памяти для предоставления приложению-потребителю информации о выбранных данных, если упомянутая совокупность пространств имен, связанных с выбранными данными, идентифицирует по меньшей мере один тип данных, связанный с выбранными данными, то предоставляют упомянутую совокупность пространств имен приложению-потребителю, выбирают по меньшей мере один тип данных для вставки выбранных данных во второй документ, анализируют список доступных пространств имен на предмет наличия ресурсов, связанных с упомянутой совокупностью пространств имен, связанных с выбранными данными, и, если данный список доступных пространств имен идентифицирует ресурс, связанный с по меньшей мере одним из упомянутой совокупности пространств имен, связанных с выбранными данными, то предоставляют этот первый ресурс приложению-потребителю, и получают первый ресурс, связанный с выбранным типом данных, для подготовки данных к вставке во второй документ посредством приложения-потребителя.

2. Способ по п.1, в котором, если список доступных пространств имен не идентифицирует первый ресурс, связанный с упомянутым по меньшей мере одним из упомянутой совокупности пространств имен, связанных с выбранными данными, то анализируют библиотеку пространств имен на предмет наличия первого ресурса, связанного с упомянутым по меньшей мере одним из упомянутой совокупности пространств имен, связанных с выбранными данными, и, если библиотека пространств имен идентифицирует первый ресурс, связанный с упомянутым по меньшей мере одним из упомянутой совокупности пространств имен, связанных с выбранными данными, то предоставляют первый ресурс приложению-потребителю.

3. Способ по п.2, в котором, если список доступных пространств имен идентифицирует второй ресурс, связанный с по меньшей мере одним другим из упомянутой совокупности пространств имен, связанных с выбранными данными, то получают этот второй ресурс, связанный с упомянутым по меньшей мере одним другим из упомянутой совокупности

пространств имен, для подготовки данных к вставке во второй документ посредством приложения-потребителя.

4. Способ по п.3, дополнительно содержащий этап, на котором предоставляют пользователю возможность выбора между вставкой выбранных данных во второй документ согласно первому ресурсу и вставкой выбранных данных во второй документ согласно второму ресурсу.

5. Способ по п.3, дополнительно содержащий этап, на котором предоставляют пользователю возможность выбора между вставкой выбранных данных во второй документ согласно по меньшей мере одному ресурсу, выбранному из одного из совокупности доступных пространств имен, причем каждое из этой совокупности доступных пространств имен связано с по меньшей мере одним ресурсом.

6. Способ по п.4, в котором при сохранении выбранных данных в памяти выбранные данные сохраняют в формате расширяемого языка разметки (XML).

7. Способ по п.6, в котором первый и второй ресурсы включают в себя преобразования расширяемого языка стилей (XSLT) для преобразования выбранных данных из первого типа данных, связанного с первым документом, во второй тип данных для вставки выбранных данных во второй документ.

8. Способ по п.7, в котором, если с выбранным типом данных не связано никаких ресурсов для подготовки данных к вставке во второй документ посредством приложения-потребителя, то вставляют выбранные данные во второй документ без использования расширенных функциональных возможностей вставки.

9. Способ по п.8, в котором при сохранении данных, связанных с выбранными данными, в области памяти, сохраняют данные в области памяти в формате CF_XML_HEADER.

10. Способ по п.9, в котором, если упомянутая совокупность пространств имен идентифицирована как связанная с первым документом, то обеспечивают совместно с упомянутой совокупностью пространств имен, связанных с выбранными данными, информацию, позволяющую приложению-потребителю определять местоположение упомянутой совокупности пространств имен в списке доступных пространств имен или библиотеке пространств имен.

11. Способ по п.10, в котором при обеспечении совместно с упомянутой совокупностью пространств имен, связанных с выбранными данными, информации, позволяющей приложению-потребителю определять местоположение упомянутой совокупности пространств имен в списке доступных пространств имен или библиотеке пространств имен, обеспечивают идентификатор версии для упомянутой совокупности пространств имен, обеспечивают универсальный указатель ресурса для упомянутой совокупности пространств имен и обеспечивают размер файла для упомянутой совокупности пространств имен.

12. Способ по п.9, в котором, если упомянутая совокупность пространств имен идентифицирована как связанная с первым документом, то предоставляют приложению-потребителю любые типы данных, связанные с первым документом, согласно упомянутой совокупности пространств имен.

13. Способ по п.12, в котором типы данных, связанные с первым документом согласно упомянутой совокупности пространств имен, предоставляют приложению-потребителю посредством формата CF_XML_VIEW.

14. Способ по п.1, в котором приложение-поставщик является приложением текстового редактора.

15. Способ по п.1, в котором приложение-поставщик является приложением электронной таблицы.

16. Способ по п.1, в котором приложение-поставщик является приложением слайдовой презентации.

17. Способ по п.1, в котором приложение-потребитель является приложением текстового редактора.

18. Способ по п.1, в котором приложение-потребитель является приложением

электронной таблицы.

19. Способ по п.1, в котором приложение-потребитель является приложением слайдовой презентации.

20. Способ вставки данных во второй документ, содержащий этапы, на которых
5 открывают первый документ посредством приложения-поставщика, применяют структуру к первому документу согласно языку разметки, выбирают данные из первого документа для вставки во второй документ, выбирают совокупность пространств имен, доступных для приложения-поставщика, связывают эту совокупность пространств имен с выбранными данными из приложения-поставщика, сохраняют выбранные данные в области памяти,
10 сохраняют упомянутую совокупность пространств имен, связанных с выбранными данными, в упомянутой области памяти для предоставления приложению-потребителю информации о выбранных данных, открывают второй документ посредством приложения-потребителя и выбирают функцию вставки приложения-потребителя для вставки выбранных данных во второй документ, считывают посредством приложения-потребителя упомянутую
15 совокупность пространств имен, связанных с выбранными данными, в ответ на считывание приложением-потребителем упомянутой совокупности пространств имен, связанных с выбранными данными, определяют, задает ли по меньшей мере одно из упомянутой совокупности пространств имен, связанных с выбранными данными, допустимое содержимое данных, тип данных и структуру данных, для структуры, примененной к
20 выбранным данным, определяют, содержит ли список доступных пространств имен нужное пространство имен из упомянутого по меньшей мере одного пространства имен для использования приложением-потребителем, если список доступных пространств имен содержит нужное пространство имен из упомянутого по меньшей мере одного пространства имен для использования приложением-потребителем, то выбирают это нужное
25 пространство имен, предоставляют выбранное пространство имен приложению-потребителю, получают посредством приложения-потребителя ресурс, связанный с выбранным пространством имен, для вставки выбранных данных во второй документ и вставляют выбранные данные во второй документ согласно данному ресурсу.

21. Способ по п.20, в котором, если список доступных пространств имен не содержит
30 нужного пространства имен из упомянутого по меньшей мере одного пространства имен для использования приложением-потребителем, то определяют, содержит ли библиотека пространств имен по меньшей мере один ресурс для упомянутого по меньшей мере одного пространства имен для использования приложением-потребителем, если библиотека пространств имен содержит по меньшей мере один нужный ресурс для
35 упомянутого по меньшей мере одного пространства имен для использования приложением-потребителем, то выбирают нужный ресурс из библиотеки пространств имен и предоставляют выбранное пространство имен и ресурс приложению-потребителю.

22. Способ по п.21, дополнительно содержащий этап, на котором предоставляют
40 пользователю возможность выбора вставки выбранных данных во второй документ согласно выбранному пространству имен.

23. Способ по п.22, в котором язык разметки является расширяемым языком разметки (XML).

24. Способ по п.23, в котором ресурс является файлом преобразования расширяемого
45 языка стилей для преобразования выбранных данных из первого типа данных, связанного с первым документом, во второй тип данных для вставки выбранных данных во второй документ.

25. Способ по п.20, в котором при сохранении упомянутой совокупности пространств имен, связанных с выбранными данными, в области памяти для предоставления приложению-потребителю информации о выбранных данных, определяют, связано ли с
50 первым документом какое-либо из упомянутой совокупности пространств имен.

26. Способ по п.25, в котором при определении того, связано ли с первым документом какое-либо из упомянутой совокупности пространств имен, предоставляют приложению-потребителю информацию для определения местоположения упомянутой совокупности

пространств имен в списке доступных пространств имен или библиотеке пространств имен.

27. Машиночитаемый носитель, на котором хранятся машиноисполняемые команды для вставки данных во второй документ, которые при исполнении на компьютере осуществляют этапы, на которых открывают первый документ посредством приложения-поставщика, 5 применяют структуру к первому документу согласно языку разметки, выбирают данные из первого документа для вставки во второй документ, выбирают совокупность пространств имен, доступных для приложения-поставщика, связывают эту совокупность пространств имен с выбранными данными из приложения-поставщика, сохраняют выбранные данные в области памяти, сохраняют упомянутую совокупность пространств имен, связанных с 10 выбранными данными, в упомянутой области памяти для предоставления приложению-потребителю информации о выбранных данных, открывают второй документ посредством приложения-потребителя и выбирают функцию вставки приложения-потребителя для вставки выбранных данных во второй документ, считывают посредством приложения-потребителя упомянутую совокупность пространств имен, связанных с выбранными 15 данными, в ответ на считывание приложением-потребителем упомянутой совокупности пространств имен, связанных с выбранными данными, определяют, задает ли по меньшей мере одно из упомянутой совокупности пространств имен, связанных с выбранными данными, допустимое содержимое данных, тип данных и структуру данных, для структуры, примененной к выбранным данным, определяют, содержит ли список доступных 20 пространств имен нужное пространство имен из упомянутого по меньшей мере одного пространства имен для использования приложением-потребителем, если список доступных пространств имен содержит нужное пространство имен из упомянутого по меньшей мере одного пространства имен для использования приложением-потребителем, то выбирают это нужное пространство имен, предоставляют выбранное пространство имен приложению- 25 потребителю, получают посредством приложения-потребителя ресурс, связанный с выбранным пространством имен, для вставки выбранных данных во второй документ и вставляют выбранные данные во второй документ согласно данному ресурсу.

28. Машиночитаемый носитель по п.27, в котором, если список доступных пространств имен не содержит нужного пространства имен из упомянутого по меньшей мере одного 30 пространства имен для использования приложением-потребителем, то определяют, содержит ли библиотека пространств имен нужное пространство имен из упомянутого по меньшей мере одного пространства имен для использования приложением-потребителем, если библиотека пространств имен содержит по меньшей мере один нужный ресурс для упомянутого по меньшей мере одного пространства имен для использования приложением- 35 потребителем, то выбирают этот нужный ресурс из библиотеки пространств имен и предоставляют выбранное пространство имен и ресурс приложению-потребителю.

29. Машиночитаемый носитель по п.28, дополнительно содержащий предоставление пользователю возможности выбора вставки выбранных данных во второй документ согласно выбранному пространству имен.

40 30. Машиночитаемый носитель по п.29, в котором язык разметки является расширяемым языком разметки (XML).

31. Машиночитаемый носитель по п.30, в котором при сохранении упомянутой совокупности пространств имен, связанных с выбранными данными, в области памяти для предоставления приложению-потребителю информации о выбранных данных, определяют, 45 связано ли с первым документом какое-либо из упомянутой совокупности пространств имен.

32. Машиночитаемый носитель по п.31, в котором при определении того, связано ли с первым документом какое-либо из упомянутой совокупности пространств имен, предоставляют приложению-потребителю информацию для определения местоположения 50 упомянутой совокупности пространств имен в списке доступных пространств имен или библиотеке пространств имен.

33. Способ вставки данных из приложения, осуществляющего копирование, в приложение-потребитель, содержащий этапы, на которых применяют первый элемент

расширяемого языка разметки (XML) к началу области первого документа, причем эту область выбирают для копирования из приложения, осуществляющего копирование, для вставки в приложение-потребитель, применяют второй элемент XML к концу выбранной области, обеспечивают в заголовке информацию, связанную с выбранной областью,

5 причем данная информация содержит идентификационные данные совокупности доступных пространств имен XML и указатели на идентификационные данные формата буфера обмена, который соответствует каждому из упомянутой совокупности доступных пространств имен, обеспечивают в заголовке информацию о размере файла для каждого из доступных пространств имен и подают список из упомянутой совокупности доступных
10 пространств имен в буфер обмена, чтобы приложение-потребитель могло выбирать на основе обеспеченной в заголовке информации из упомянутой совокупности доступных пространств имен для вставки данных из приложения, осуществляющего копирование.

15

20

25

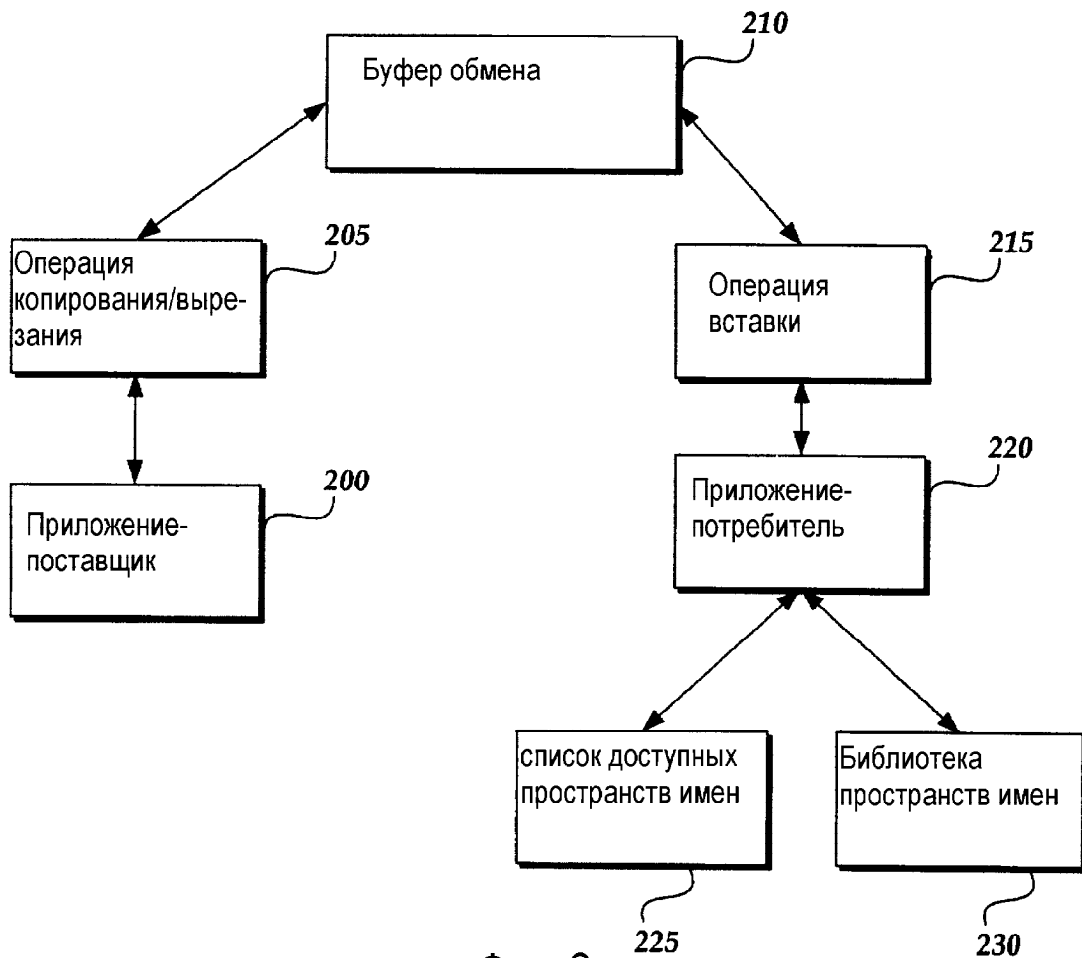
30

35

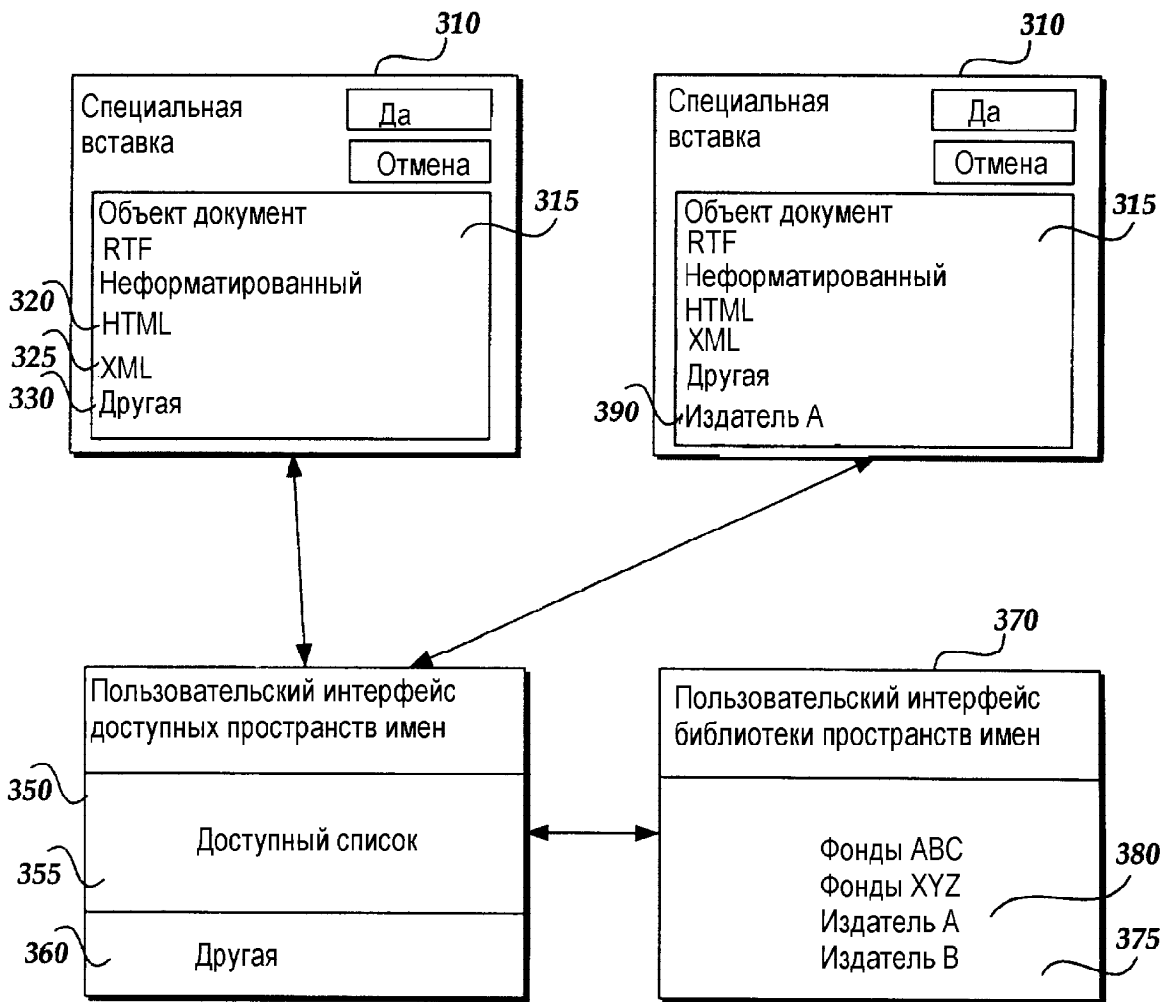
40

45

50



Фиг. 2



Фиг. 3

400

Selling_Memo.doc - Microsoft Word

Heading 1A Arial 11

Final Showing Markup

1 2 3 4 5 6

410 ("Base Plan") financial statements and management's discussion and analysis are included in Section VIII. A number of other forecasted/unbudgeted opportunities are highlighted in Section V. When these unbudgeted opportunities are considered into account, sales and EBITDA are expected to grow at annual rates of 13.1% and 15.6%, respectively, from 1997-2001.

Table 1 Eight Months Ended August 31

\$ millions	1996		Growth (%)
	(\$)	% of Sales	
Sales	89.8	—	19.3
Gross Profit	36.8	40.9	17.1
EBITDA	23.4	26.1	17.9
EBITA	21.1	23.5	16.6
Cash Flow Components			
Capital Expenditures	3.2		
Net Working Capital	17.1		

415

420

425

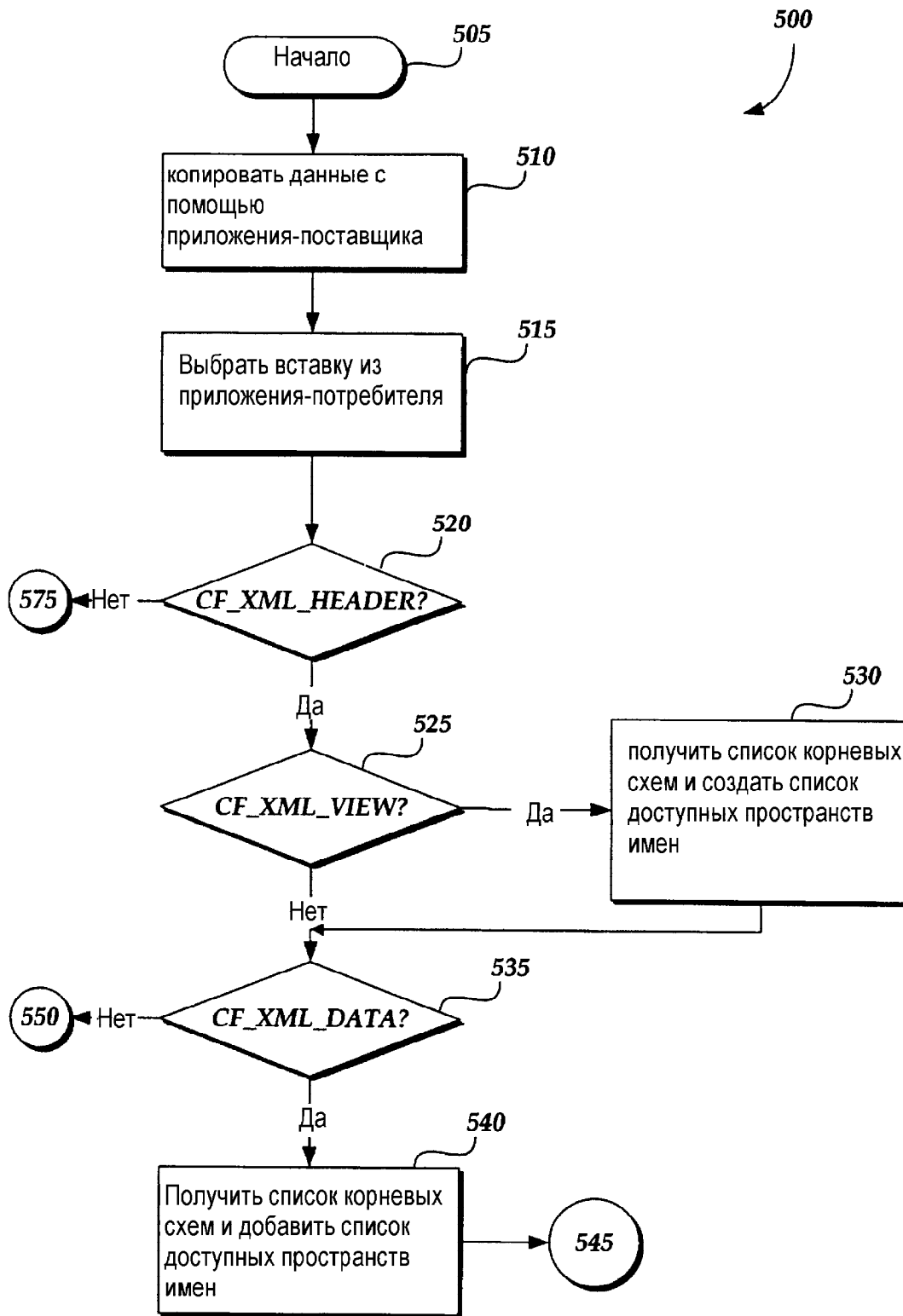
430

435

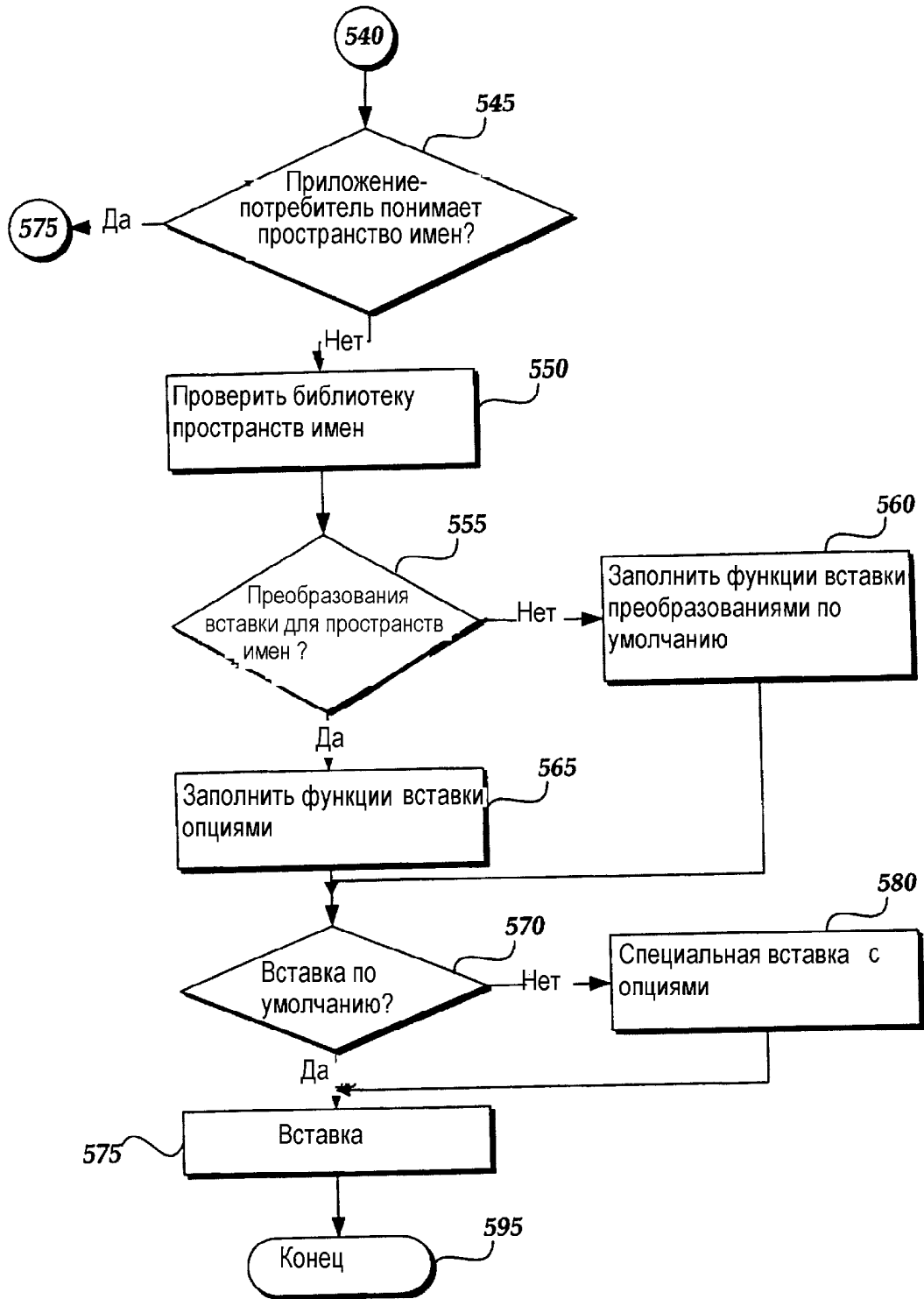
440

- Eight Month History
- Projected Financial Summary
- Industry Breakdown
- Apply Style or Formatting...

Фиг. 4



Фиг. 5



Фиг. 6