



ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ,
ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21), (22) Заявка: **2005119663/09**, **30.07.2004**

(24) Дата начала отсчета срока действия патента:
30.07.2004

(30) Конвенционный приоритет:
23.10.2003 US 10/691,888

(43) Дата публикации заявки: **20.01.2006**

(45) Опубликовано: **10.06.2009** Бюл. № 16

(56) Список документов, цитированных в отчете о
поиске: **RU 2210809 C2**, **20.08.2003**. **RU 12619 U1**,
20.01.2000. **US 6008806 A**, **28.12.1999**. **US**
6101513 A, **08.08.2000**. **US 6121964 A**,
19.09.2000.

(85) Дата перевода заявки РСТ на национальную
фазу: **22.06.2005**

(86) Заявка РСТ:
US 2004/025306 (30.07.2004)

(87) Публикация РСТ:
WO 2005/045592 (19.05.2005)

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):

МАККИ Тимоти П. (US),
ДЕ ВОРЧИК Дэвид Джордж (US),
ШЕЛДОН Дэвид Джоуэл (US)

(73) Патентообладатель(и):

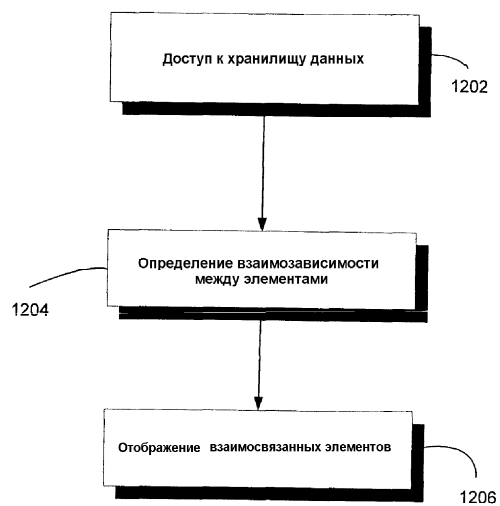
МАЙКРОСОФТ КОРПОРЕЙШН (US)

(54) СИСТЕМА И СПОСОБ ПРЕДСТАВЛЕНИЯ ДЛЯ ПОЛЬЗОВАТЕЛЯ ВЗАИМОСВЯЗАННЫХ ЭЛЕМЕНТОВ

(57) Реферат:

Изобретение относится к средствам отображения элементов, хранящихся в компьютере. Техническим результатом является обеспечение сред для отображения элементов с соответствующей контекстной информацией и взаимосвязанных с этими элементами задачами. В системе и способе обеспечивают представление для пользователя взаимосвязанных элементов универсального хранилища данных. Часть элементов содержит

реляционную информацию, которая позволяет определять взаимозависимость между двумя или более из множества элементов. Оболочка представляет выбранный элемент пользователю и сконфигурирована для использования реляционной информации, для представления одного или больше элементов в хранилище данных, которые взаимосвязаны с выбранным элементом. 4 н. и 21 з.п. ф-лы, 12 ил.



Фиг.12 1200



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY,
PATENTS AND TRADEMARKS

(12) ABSTRACT OF INVENTION

(21), (22) Application: **2005119663/09, 30.07.2004**

(24) Effective date for property rights:
30.07.2004

(30) Priority:
23.10.2003 US 10/691,888

(43) Application published: **20.01.2006**

(45) Date of publication: **10.06.2009 Bull. 16**

(85) Commencement of national phase: **22.06.2005**

(86) PCT application:
US 2004/025306 (30.07.2004)

(87) PCT publication:
WO 2005/045592 (19.05.2005)

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):

**MAKKI Timoti P. (US),
DE VORChIK Dehvid Dzhordzh (US),
ShELDON Dehvid Dzhouehl (US)**

(73) Proprietor(s):

MAJKROSOFT KORPOREJShN (US)

(54) SYSTEM AND METHOD OF PRESENTING INTERRELATED COMPONENTS TO USERS

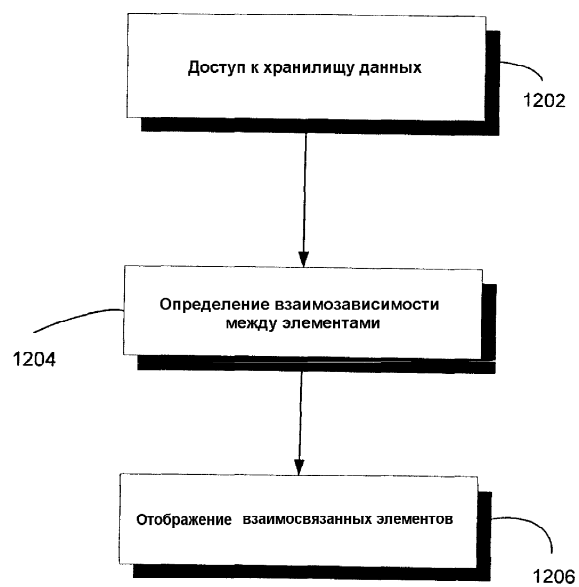
(57) Abstract:

FIELD: physics; computer engineering.

SUBSTANCE: present invention relates to means of displaying elements, stored on a computer. In the system and method there is presentation of interrelated elements of a universal data bank to a user. Part of the elements contains relational information, which allows for determining interdependency between two or more elements. A shell presents the element chosen by the user and is configured to use the relational information, for presenting one or more elements in the data bank, which are interrelated with the chosen element.

EFFECT: provision for means of displaying elements with the corresponding context information and tasks interrelated with these elements.

25 cl, 13 dwg



Фиг.12 1200

Область техники

Настоящее изобретение, в целом, относится к области компьютерного программного обеспечения. Более конкретно, настоящее изобретение относится к системе и способу, предназначенное для отображения для пользователя элементов, хранящихся в компьютере.

Предшествующий уровень техники

Обеспечение пользователей компьютеров возможностью быстро находить и отображать части информации, независимо от формата информации или места ее расположения, представляет собой проблему, которую в течение многих лет пытаются решить в компьютерной промышленности. В настоящее время эта проблема стоит более остро, чем когда-либо, поскольку все большее количество людей используют компьютеры в своей повседневной деятельности, и в компьютерах сохраняют все большее количество информации разного рода.

Традиционно, как в операционной системе WINDOWS® 98™, компании Microsoft Corporation, такая информация сохраняется в компьютере в информационном хранилище с использованием иерархической структуры, организованной в виде файлов информации или на носителе, в папках. Хотя этот способ организации информационного хранилища широко используют в течение многих лет, он имеет ограничения, связанные с тем, что некоторые данные расположены за пределами иерархии файлов, и при этом пользователи ограничены форматом и расположением при поиске требуемых элементов информации. Для решения этой проблемы производители компьютерного программного обеспечения в настоящее время работают над новыми схемами организации информационного хранилища, которые могли бы составить альтернативу традиционной иерархической файловой структуре.

Пример такого альтернативного информационного хранилища раскрыт в находящейся в совместной собственности, одновременно находящейся на рассмотрении заявке "SYSTEM AND METHODS FOR REPRESENTING UNITS OF INFORMATION MANAGEABLE BY A HARDWARE/SOFTWARE INTERFACE SYSTEM BUT INDEPENDENT OF PHYSICAL REPRESENTATION", заявка на американский патент номер (еще назначен) (регистрационный номер 7 патентного поверенного № MSFT-1748). Эта одновременно находящаяся на рассмотрении заявка была подана 21 августа 2003, и в ней описано информационное хранилище, которое объединяет записанные данные в единую базу данных. Эта база данных представляет собой одно место, где хранятся все данные; при этом существует только один способ представления данных в базе данных и только один способ запроса данных. Благодаря замене устаревшей файловой системы этой современной технологией базы данных, обеспечивается простота поиска в информационном хранилище, более высокая надежность, лучший доступ и более высокая устойчивость.

При наличии такого объединенного информационного хранилища возникает потребность обеспечения пользователей соответствующими инструментальными средствами и возможностями взаимодействия с сохраненными данными. Обычная операционная система, такая как WINDOWS® 2000™, компании Microsoft Corporation, содержит сервисную программу (утилиту) оболочки, которая обеспечивает для пользователя интерфейс просмотра различной информации о компьютере. Такая оболочка обычно включает в себя программу просмотра файловой системы, которая позволяет пользователю перемещаться по файловой системе и находить и открывать файлы и папки. Например, WINDOWS® EXPLORER™ компании Microsoft Corporation представляет собой сервисную программу просмотра файловой системы, включенную

в операционную систему WINDOWS® 2000™.

Эта оболочка также дает пользователям возможность просматривать элементы, не являющиеся файлами, такие как принтеры или шрифты. Возможность навигации обеспечивается, благодаря тому, что типовая оболочка запрограммирована с определенными функциональными возможностями отображения таких специальных элементов, как если бы они были расположены в файловой системе. Например, в WINDOWS® 2000™ пользователь может открыть папку "Принтеры", расположенную в пункте "Панель управления" меню "Пуск". Поскольку принтеры являются элементами аппаратных средств, а не файлами, графическое представление принтеров выполняется с использованием специальной программы, предназначенной для отображения принтеров, как если бы они были файлами, постоянно находящимися в папке "Принтеры". Однако использование специальной (заказной) программы и специальных (сделанных на заказ) изображений представляет сложность для разработчиков программного обеспечения, может быть ненадежным и снижает устойчивость программы просмотра оболочки. Кроме того, когда эта специальная программа или специальные изображения находятся не на месте, предназначенном для данных такого типа, оболочка будет не способна отображать элементы этого типа. В соответствии с этим обычные оболочки ограничены по своим возможностям и гибкости при отображении некоторых элементов для пользователя.

Другое ограничение обычных программ просмотра - оболочек состоит в их ограниченной способности отображения взаимосвязанных элементов. Обычно программа просмотра - оболочка позволяет отображать элементы только в иерархической структуре, в которой они сохранены и организованы, в виде файлов, сохраненных в папках. Например, если пользователю требуется рассмотреть все файлы - изображения, сохраненные в компьютере, пользователь должен сначала поместить все такие файлы - изображения в одну папку. Поскольку оболочка имеет ограниченные возможности по определению взаимоотношения между элементами, пользователю трудно просматривать файлы в контексте их взаимосвязи.

Кроме того, обычные программы просмотра - оболочки ограничены по способности отображения наборов элементов в пределах контекстуально приспособленной (организованной) среды, которая сопоставляет взаимосопоставляющую информацию и задачи с набором отображаемых элементов. Разработчики, благодаря использованию такого сопоставления, обеспечивают пользователей возможностью представления соответствующей информации и инструментальных средств, требуемых для навигации между элементами, улучшают возможность выполнения общих задач, ассоциированных с этими элементами. Известный уровень техники, однако, не позволяет разработчикам получить такой результат без использования специальных программ.

Пример представления файлов в расширенной среде с использованием специальной программы может быть представлен папкой "Мои рисунки", которая включена в операционную систему WINDOWS® XP™ компании Microsoft Corporation. Когда файлы изображения сохранены в папке "Мои рисунки", пользователь может просматривать изображения с различным размером, поворачивать их, рассматривать последовательность рисунков, печатать изображения или копировать изображения на компакт-диск. В оболочке в WINDOWS® XP™ используется специальная программа, предназначенная для включения этих задач, связанных с изображением, в отображение этой папки так, что для пользователя, при сохранении изображений в этой специальной папке, обеспечивается возможность простой навигации среди

изображений и выполнения общих задач с файлами. Однако в этой среде отображаются только файлы, сохраненные в папке "Мои рисунки", и для обеспечения этих функциональных возможностей используется специальная программа. В то время как папка "Мои рисунки" представляет собой усовершенствование по сравнению с традиционным представлением элементов, возможности разработчиков все еще ограничены при определении таких сред с насыщенным содержанием, без использования специальных программ.

В соответствии с этим существует потребность в улучшенной оболочке, которая позволила бы отображать каждый элемент, находящийся в универсальном информационном хранилище, и кроме того, существует потребность в улучшенной оболочке, сконфигурированной для представления элементов в универсальном информационном хранилище в контексте возникновения их соотношений. Также существует потребность в улучшенных возможностях оболочки для разработчиков по созданию специальных сред, отображающих элементы с соответствующей контекстной информацией и взаимосвязанными с ними задачами, без необходимости использования специальных программ.

Сущность изобретения

Настоящее изобретение обеспечивает выполнение вышеуказанных задач и преодоление одного или больше недостатков известного уровня техники с помощью системы, сконфигурированной для представления для пользователя взаимосвязанных элементов, хранящихся в информационном хранилище. В одном аспекте настоящего изобретения предложена система, предназначенная для представления для пользователя взаимосвязанных элементов. В систему включено универсальное информационное хранилище, которое содержит множество элементов. По меньшей мере, часть элементов в информационном хранилище содержит реляционную (связанную, родственную) информацию. В оболочке реляционную информацию используют для представления взаимосвязанных элементов.

В другом аспекте настоящее изобретение предлагает способ, реализованный на компьютере, предназначенный для представления для пользователя взаимосвязанных элементов. С помощью этого способа обеспечивается доступ к данным в универсальном информационном хранилище, которое содержит элементы, включающие в себя информацию, позволяющую определить соотношения между элементами в информационном хранилище. Такая реляционная информация используется для определения взаимосвязи (соотношения) между элементами, и взаимосвязанные элементы отображаются для пользователя.

Краткое описание чертежей

Настоящее изобретение подробно описано ниже со ссылками на прилагаемые чертежи, на которых:

На фиг.1 показана блок-схема среды компьютерной системы, пригодной для использования при выполнении настоящего изобретения;

На фиг.2А показана блок-схема, иллюстрирующая компьютерную систему, разделенную на три группы компонентов: компонент аппаратных средств, компонент системы интерфейса аппаратных/программных средств и компонент прикладных программ;

на фиг.2В представлена традиционная древовидная иерархическая структура файлов, сгруппированных в папки, в директории (каталоге) в операционной системе на основе файлов;

на фиг.3 показана блок-схема, иллюстрирующая платформу хранилища, в

соответствии с настоящим изобретением;

на фиг.4 представлена структурная взаимозависимость между элементами, папками элементов и категориями в различных вариантах выполнения настоящего изобретения;

на фиг.5 показана схема данных, содержащихся в одном элементе, в соответствии с вариантом выполнения настоящего изобретения;

на фиг.6 показана схема последовательности выполнения операций, изображающая способ представления для пользователя одного или больше элементов, в соответствии с одним вариантом выполнения настоящего изобретения;

на фиг.7 показана схема последовательности выполнения операций, изображающая способ представления одного или больше элементов для пользователя, в соответствии с одним вариантом выполнения настоящего изобретения;

на фиг.8 показана схема, изображающая иерархию схемы просмотра, в соответствии с вариантом выполнения настоящего изобретения;

на фиг.9 показана схема, изображающая иерархию схемы просмотра, включающая схему просмотра программы - "проводника", в соответствии с вариантом выполнения настоящего изобретения;

на фиг.10 показана схема последовательности выполнения операций, представляющая способ представления элементов для пользователя, в соответствии с вариантом выполнения настоящего изобретения; и

на фиг.11 показана схема последовательности выполнения операций, представляющая способ представления для пользователя взаимосвязанных элементов, в соответствии с вариантом выполнения настоящего изобретения.

Подробное описание изобретения

I. ВВЕДЕНИЕ

Объект настоящего изобретения описан со спецификой, необходимой для удовлетворения требований, предусмотренных патентным законодательством.

Однако само описание не предназначено для ограничения объема данного патента. Скорее авторы настоящего изобретения предусматривают, что заявленный объект также может быть выполнен с использованием других способов, предусматривающих включение различных этапов или комбинации этапов, аналогичных описанным в данном документе, совместно с другими существующими или будущими технологиями. Кроме того, хотя термин "этап" может использоваться здесь для обозначения различных элементов используемых способов, этот термин не следует интерпретировать как предполагающий какой-либо конкретный порядок среди или между различными описанными здесь этапами, если только не и за исключением случая, когда порядок отдельных этапов будет явно описан.

Настоящее изобретение раскрывает усовершенствованную систему и способ отображения для пользователя элементов, хранящихся в компьютере. Ниже описан пример операционной среды для настоящего изобретения.

A. ПРИМЕР ОПЕРАЦИОННОЙ СРЕДЫ

Различные варианты выполнения настоящего изобретения могут быть выполнены на компьютере. Фиг.1 и приведенное ниже описание предназначены для представления короткого общего описания соответствующей компьютерной среды, в которой может быть выполнено настоящее изобретение. Хотя это не требуется, различные аспекты изобретения могут быть описаны в общем контексте выполняемых на компьютере инструкций, таких как программные модули, выполняемые компьютером, таком как рабочая станция - клиент или сервер. Обычно программные модули включают в себя процедуры, программы, объекты,

компоненты, структуры данных и т.п., которые выполняют определенные задачи или реализуют различные конкретные абстрактные типы данных. Кроме того, настоящее изобретение может быть использовано на практике с другими конфигурациями компьютерной системы, включая портативные устройства, многопроцессорные системы, микропроцессор на основе программируемых электронных устройств потребителя, сетевые ПК, микрокомпьютеры, главные компьютеры и т.п. Настоящее изобретение также можно использовать на практике в распределенных компьютерных средах, в которых задачи выполняют с использованием удаленных устройств обработки информации, соединенных через сеть передачи данных. В распределенной компьютерной среде программные модули могут быть расположены как на локальных, так и на удаленных устройствах хранения информации.

Как показано на фиг.1, пример компьютерной системы общего назначения включает в себя обычный персональный компьютер 20 или подобное устройство, включающее в себя блок 21 процессора, системное запоминающее устройство 22 и системную шину 23, которая подсоединяет различные компоненты системы, включая системное запоминающее устройство, к блоку 21 процессора. Системная шина 23 может представлять собой любую структуру из нескольких типов шин, включая шину памяти или контроллер памяти, периферийную шину и локальную шину с использованием любой из множества различных архитектур шины. Системное запоминающее устройство включает в себя постоянное запоминающее устройство (ПЗУ) 24 и оперативное запоминающее устройство (ОЗУ) 25. Базовая система ввода-вывода (БИОС, BIOS), содержащая основные процедуры, которые обеспечивают возможность передачи информации между элементами в персональном компьютере 20, например, во время включения, записана в ПЗУ 24. Персональный компьютер 20, кроме того, может содержать привод 27 жесткого диска, предназначенный для считывания и записи информации на жесткий диск, не показан, привод 28 магнитного диска для считывания с или записи информации на съемный магнитный диск 29, и привод 30 оптического диска, предназначенный для считывания или записи информации на съемный оптический диск 31, такой как CD ROM (постоянное запоминающее устройство на компакт-диске) или другой оптический носитель. Привод 27 жесткого диска, привод 28 магнитного диска и привод 30 оптического диска соединены с системной шиной 23 с помощью интерфейса 32 привода жесткого диска, интерфейса 33 привода магнитного диска и интерфейса 34 привода оптического диска, соответственно. Приводы и ассоциированные с ними считываемые компьютером носители обеспечивают энергонезависимое хранилище считываемых компьютером инструкций, структур данных, программных модулей и других данных для персонального компьютера 20. Хотя в описанном здесь примере структуры используется жесткий диск, съемный магнитный диск 29 и съемный оптический диск 31, для специалистов в данной области техники понятно, что другие типы считываемых компьютером носителей, на которые можно записывать данные, доступ к которым обеспечивается с помощью компьютера, такие как магнитные кассеты, карты запоминающего устройства типа флэш, цифровые видеодиски, картриджи Бернулли, оперативные запоминающие устройства (ОЗУ), постоянные запоминающие устройства (ПЗУ) и т.п., также можно использовать в примере рабочей среды. Аналогично, в пример среды также можно включить множество типов устройств контроля, таких как тепловые датчики и системы безопасности или пожарной тревоги, и другие источники информации.

Множество программных модулей может быть записано на жесткий диск,

магнитный диск 29, оптический диск 31, ПЗУ 24 или ОЗУ 25, включая операционную систему 35, одну или больше прикладных программ 36, других программных модулей 37 и программных данных 38. Пользователь может вводить команды и информацию в персональный компьютер 20 через устройства ввода, такие как клавиатура 40 и устройство 42 - указатель. Другие устройства ввода (не показаны) могут включать в себя микрофон, джойстик, игровой планшет, дополнительный диск, сканер или тому подобное. Эти и другие устройства ввода часто подключены к блоку 21 процессора через интерфейс 46 последовательного порта, который соединен с системной шиной, но могут быть соединены с помощью других интерфейсов, таких как параллельный порт, игровой порт или универсальная последовательная шина (УПШ, USB). Монитор 47 или устройство отображения другого типа также соединен с системной шиной 23 через интерфейс, такой как видеоадаптер 48. Кроме монитора 47, персональные компьютеры обычно включают в себя другие периферийные выходные устройства (не показаны), такие как громкоговорители и принтеры. Пример системы, показанной на фиг.1, также включает адаптер 55 с главным компьютером, шину 56 интерфейса малых вычислительных систем (ИМВС, SCSI) и внешнее устройство 62 хранения, подключенное к шине 56 SCSI.

Персональный компьютер 20 может работать в сетевой среде с использованием логических соединений с одним или больше удаленных компьютеров, таких как удаленный компьютер 49. Удаленный компьютер 49 может представлять собой другой персональный компьютер, сервер, маршрутизатор, сетевой ПК, одноранговое устройство или другой узел общей сети, и обычно включает в себя множество или все элементы, описанные выше в отношении персонального компьютера 20, хотя на фиг.1 изображено только запоминающее устройство 50. Логические соединения, представленные на фиг.1, включают в себя локальную сеть (LAN) 51 и глобальную сеть (WAN) 52. Такие сетевые среды являются общеупотребимыми в офисах, компьютерных сетях предприятия, сетях Интранет и Интернет.

При использовании в среде локальной сети ЛВС (LAN) персональный компьютер 20 соединен с LAN 51 через сетевой интерфейс или адаптер 53. При использовании в сетевой среде глобальной вычислительной сети ГВС (WAN) персональный компьютер 20 обычно включает в себя модем 54 или другие средства установления связи через глобальную сеть 52, такую как Интернет. Модем 54, который может быть внутренним или внешним, соединен с системной шиной 23 через интерфейс 46 последовательного порта. В сетевой среде программные модули, изображенные как относящиеся к персональному компьютеру 20 или его частям, могут быть сохранены в удаленном устройстве хранения информации. Следует понимать, что показанные сетевые связи представляют собой только пример, и могут использоваться другие средства установления каналов передачи данных между компьютерами.

Как показано в блок-схеме на фиг.2А, компьютерная система 200 может быть приблизительно разделена на три группы компонентов: компонент 202 аппаратных средств, компонент 204 системы интерфейса аппаратных/программных средств и компонент 206 прикладных программ (также называемый "компонентом пользователя" или "программным компонентом" в некотором контексте приведенного здесь описания).

В различных вариантах выполнения компьютерной системы 200, как снова показано на фиг.1, компонент 202 аппаратных средств может содержать центральное процессорное устройство (ЦПУ) 21, запоминающее устройство (ПЗУ 24 и ОЗУ 25),

базовую систему 26 входа-выхода (BIOS) и различные устройства ввода-вывода (I/O), такие как клавиатура 40, мышь 42, монитор 47 и/или принтер (не показан), помимо других элементов. Компонент 202 аппаратных средств содержит основную (базовую) физическую инфраструктуру компьютерной системы 200.

Компонент 206 прикладных программ содержит различные программы, включая, без ограничения компиляторы, системы баз данных, текстовые редакторы, программы для выполнения деловых операций, видеоигры и т. д. Прикладные программы обеспечивают средство, с помощью которого используются компьютерные ресурсы для решения задач, получают решения и обрабатывают данные для различных пользователей (машин, других компьютерных систем и/или конечных пользователей).

Компонент 204 системы интерфейса аппаратных/программных средств содержит операционную систему (и в некоторых вариантах выполнения может состоять исключительно из операционной системы), которая сама по себе в большинстве случаев содержит оболочку и ядро. "Операционная система" (ОС, OS) представляет собой специальную программу, которая действует как посредник между прикладными программами и аппаратными средствами компьютера. Компонент 204 системы интерфейса аппаратных/программных средств также может содержать администратор виртуальной машины (ABM, VMM), общезыковую исполняющую среду (ОЯИ, CLR) или ее функциональный эквивалент, виртуальную Java-машину (JVM) или ее функциональный эквивалент, или другие программные компоненты такого рода, используемые вместо или в добавление к операционной системе компьютерной системы. Назначение системы интерфейса аппаратных/программных средств состоит в создании среды, в которой пользователь может выполнять прикладные программы. Назначение любой системы интерфейса аппаратных/программных средств состоит в обеспечении удобства пользования компьютерной системой, а также в эффективном использовании компьютерных аппаратных средств.

Систему интерфейса аппаратных/программных средств обычно загружают в компьютерную систему при включении, и после этого она управляет всеми прикладными программами в компьютерной системе. Прикладные программы взаимодействуют с системой интерфейса аппаратных/программных средств путем запроса услуг через интерфейс программ приложения (ИПП, API). Некоторые программы приложения обеспечивают для конечных пользователей возможность взаимодействовать с системой интерфейса аппаратных/программных средств через интерфейс пользователя, такой как интерфейс общего языка или графический интерфейс пользователя (ГИП, GUI).

Система интерфейса аппаратных/программных средств традиционно выполняет множество услуг для прикладных программ. В многозадачной системе интерфейса аппаратных/программных средств, в которой множество программ могут выполняться одновременно, система интерфейса аппаратных/программных средств определяет, какие прикладные программы должны работать, в каком порядке, и сколько времени должно быть выделено для работы каждой прикладной программы перед переключением на другую прикладную программу. Система интерфейса аппаратных/программных средств также управляет совместным использованием внутренней памяти множеством прикладных программ и управляет вводом и выводом через подключенные аппаратные устройства, такие как жесткие диски, принтеры и порты коммутируемой линии передачи. Система интерфейса аппаратных/программных средств также передает сообщения в каждую прикладную

программу (и, в некоторых случаях, конечному пользователю) о рабочем состоянии операций и о любых ошибках, которые могут возникнуть. Система интерфейса аппаратных/программных средств также может освобождать от излишней нагрузки средства управления выполнением пакетных заданий (например, печати) так, что
 5 иницилирующая прикладная программа освобождается от этой работы и может возобновить другую обработку и/или другие операции. На компьютерах, которые позволяют обеспечить параллельную обработку информации, система интерфейса аппаратных/программных средств также управляет разделением программы так, что
 10 она работает на более чем одном процессоре одновременно.

Оболочка системы интерфейса аппаратных/программных средств (для простоты называемая здесь "оболочкой") представляет собой интерактивный интерфейс конечного пользователя с системой интерфейса аппаратных/программных средств. (Оболочка может также называться "интерпретатором команд" или в операционной
 15 системе "оболочкой операционной системы"). Оболочка представляет собой внешний уровень системы интерфейса аппаратных/программных средств, доступ к которому осуществляется непосредственно прикладными программами и/или конечными пользователями. "Оболочка - программа просмотра" обеспечивает интерфейс
 20 пользователя, позволяющий пользователю просматривать и взаимодействовать с интерфейсом аппаратных/программных средств. В отличие от оболочки ядро представляет собой самый внутренний уровень системы интерфейса аппаратных/программных средств, который взаимодействует непосредственно с аппаратными компонентами.

Хотя предусматривается, что различные варианты выполнения настоящего изобретения особенно хорошо приспособлены для компьютеризированных систем, ничто в данном документе не ограничивает настоящее изобретение такими
 25 вариантами выполнения. Напротив, используемый здесь термин "компьютерная система" предназначен для охвата любого и всех устройств, позволяющих сохранять и обрабатывать информацию и/или позволяющих использовать сохраненную
 30 информацию для управления поведением или выполнением самого устройства, независимо от того, являются ли такие устройства электронными, механическими, логическими или виртуальными по своей природе.

35 В. ТРАДИЦИОННЫЕ ХРАНИЛИЩА НА ОСНОВЕ ФАЙЛОВ

В большинстве современных компьютерных систем "файлы" представляют собой модули хранимой информации, которые могут включать в себя систему интерфейса аппаратных/программных средств, а также прикладные программы, наборы данных
 40 и т.д. Во всех современных системах интерфейса аппаратных/программных средств (Windows, Unix, Linux, Mac OS, системах виртуальной машины и т.д.) файлы являются основными дискретными (обладающими возможностью хранения и извлечения) модулями информации (например, данные, программы и т.д.), которыми можно манипулировать с помощью системы интерфейса аппаратных/программных
 45 средств. Группы файлов обычно организованы в "папки". В системе Microsoft Windows, Macintosh OS и в других системах интерфейса аппаратных/программных средств папка представляет собой набор файлов, которые могут быть извлечены, которые можно перемещать и которыми можно по-другому манипулировать, как
 50 одиночными блоками информации. Эти папки, в свою очередь, организованы в виде древовидной иерархической структуры, называемой "директорией" (каталогом) (более подробно описана ниже). В некоторых других системах интерфейса аппаратных/программных средств, таких как DOS, z/OS и большинстве операционных

систем на основе Unix, термины "директория" и/или "папка" используются как взаимозаменяемые термины, и в ранних компьютерных системах Apple (например, Apple IIe) использовался термин "каталог" вместо термина "директория";
 5 однако все эти используемые здесь термины считаются синонимами и взаимозаменяемыми терминами, и предполагается, что они содержат все другие эквивалентные термины и ссылки на иерархические структуры хранения информации, а также на их компоненты в виде папок и файлов.

Традиционно, директория (также известная как директория папок), представляет собой древовидную иерархическую структуру, в которой файлы сгруппированы в папки, и папка, в свою очередь, расположена в соответствии с относительным расположением узлов, которые содержат дерево директорий. Например, как показано на фиг.2В, основная папка 212 файловой системы на основе DOS (или "корневая директория") может содержать множество папок 214, каждая из которых может
 15 дополнительно содержать дополнительные папки (в качестве "подпапок" этой конкретной папки) 216, и каждая из них также может содержать дополнительные папки 218 и так до бесконечности. Каждая из этих папок может иметь один или более файлов 220, хотя на уровне системы интерфейса аппаратных/программных средств
 20 отдельные файлы в папке не имеют ничего общего с другими файлами, кроме их расположения в древовидной иерархии. Не удивительно, что такой подход организации файлов в виде иерархии папок опосредованно отражает физическую организацию типичной среды хранения, используемой для сохранения этих файлов (например, жесткие диски, гибкие диски, CD-ROM и т. д.).

В дополнение к вышеуказанному, каждая папка представляет собой контейнер для своих подпапок и файлов, то есть каждая папка содержит свои подпапки и файлы. Например, когда папку удаляют с помощью системы интерфейса аппаратных/программных средств, эти подпапки и файлы в этой папке также
 30 удаляют (которые, в случае каждой подпапки, дополнительно включают в себя свои собственные подпапки и файлы в виде рекурсивной последовательности). Аналогично, каждый файл обычно принадлежит только одной папке и, хотя файл может быть скопирован, и копия может быть расположена в другой папке, сама копия файла отличается от оригинала и представляет собой отличающийся и
 35 отдельный модуль, который не имеет непосредственной связи с оригиналом (например, изменение оригинального файла не отражается в копии файла на уровне системы интерфейса аппаратных/программных средств). В этом отношении файлы и папки, таким образом, являются характерно "физическими" по своей природе,
 40 поскольку папки обрабатывают как физические контейнеры, и файлы обрабатывают как дискретные и отдельные физические элементы, находящиеся внутри этих контейнеров.

II. УНИВЕРСАЛЬНОЕ ХРАНИЛИЩЕ ДАННЫХ

Платформа для хранения, используемая в настоящем изобретении, продлевает и
 45 расширяет платформу данных за пределы описанных выше существующих в настоящее время файловых систем и разработана для хранения всех типов данных. Хранилище данных, разработанное для хранения всех типов данных, можно назвать универсальным хранилищем данных. Пример универсального хранилища данных, пригодного для использования в настоящем изобретении, описан в находящейся в
 50 совместной собственности, одновременно находящейся на рассмотрении заявке "SYSTEM AND METHODS FOR REPRESENTING UNITS OF INFORMATION MANAGEABLE BY A HARDWARE/SOFTWARE INTERFACE SYSTEM BUT

INDEPENDENT OF PHISICAL REPRESENTATION", заявка на американский патент номер (еще не назначен) (регистрационный номер патентного поверенного MSFT-1748), поданной 21 августа 2003 г., которая приведена здесь в качестве ссылки.

А. ОБЗОР

Как показано на фиг.3, платформа 300 хранения, в соответствии с настоящим изобретением, содержит универсальное хранилище 302 данных, выполненное на основе механизма 314 базы данных. В одном варианте выполнения механизм 314 базы данных содержит механизм реляционной базы данных с объектными реляционными расширениями. В одном варианте выполнения механизм 314 реляционной базы данных содержит механизм реляционной базы данных Microsoft SQL Server.

Универсальное хранилище 302 данных реализует модель 304 данных, которая поддерживает организацию, поиск, совместное использование, синхронизацию и безопасность данных. Конкретные типы данных описаны в схемах, таких как схема 340, и платформа 300 хранилища обеспечивает инструментальные средства 346 для разворачивания этих схем, а также для расширения этих схем, как более полно описано ниже.

Механизм 306 отслеживания изменений, реализованный в универсальном хранилище 302 данных, обеспечивает возможность отслеживания изменений в хранилище данных. Универсальное хранилище 302 данных также обеспечивает возможности 308 безопасности и возможности 310 повышения/понижения. Универсальное хранилище 302 данных также обеспечивает набор интерфейсов 312 программ приложений, предназначенных для обеспечения предоставления возможностей универсального хранилища 302 данных для других компонентов платформы хранения и прикладных программ (например, прикладных программ 350А, 350В и 350С), в которых используется платформа хранения.

Платформа хранения, в соответствии с настоящим изобретением, кроме того, содержит интерфейсы 322 программ приложения (API), которые позволяют прикладным программам, таким как прикладные программы 350А, 350В и 350С получать доступ ко всем приведенным выше возможностям платформы хранения и получать доступ к данным, описанным в схемах. Платформа API 322 хранения может быть использована прикладными программами в комбинации с другими API, такими как OLE (связывание и встраивание объектов) DB (база данных) API 324 и Microsoft Windows Win32 API 326.

Платформа 300 хранения, в соответствии с настоящим изобретением, может предоставлять различные услуги 328 для прикладных программ, включая услугу 330 синхронизации, которая позволяет облегчить совместное использование данных для пользователей или систем. Например, услуга 330 синхронизации может обеспечить возможность взаимодействия с другими хранилищами 340 данных, имеющими тот же формат, что и хранилище 302 данных, а также доступ к хранилищам 342 данных, имеющим другие форматы. Платформа 300 хранения также обеспечивает возможности файловой системы, которые позволяют обеспечить возможность взаимодействия универсального хранилища 302 данных с существующими файловыми системами, такими как файловая система 318 NTFS (файловая система Windows NT) операционной системы Windows.

В по меньшей мере некоторых вариантах выполнения платформа 320 хранения также обеспечивает дополнительные возможности для прикладных программ по обеспечению воздействия на данные и обеспечению взаимодействия с другими системами. Эти возможности могут быть выполнены в форме дополнительных

услуг 328, таких как услуга 334 Info Agent и услуга 332 уведомления, а также в форме других служебных программ (утилит) 336.

В по меньшей мере некоторых вариантах выполнения платформа хранения выполнена в составе или в форме части единой системы интерфейса аппаратных/программных средств компьютерной системы. Например, и без ограничений, платформа хранения в соответствии с настоящим изобретением может быть выполнена в форме части единой операционной системы, администратора виртуальной машины (VMM), общезыковой исполняющей среды (CLR) или ее функционального эквивалента, или в форме виртуальной Java-машины (JVM) или ее функционального эквивалента.

Благодаря организации общей системы хранения и схематизированных данных, платформа хранения в соответствии с настоящим изобретением обеспечивает более эффективную разработку прикладных программ для потребителей, специалистов в области анализа и обработки информации. Она предлагает богатое и широкое пространство для программирования, которое не только делает доступными возможности, присущими ее модели данных, но также охватывает и расширяет существующую файловую систему и способы доступа к базе данных.

В. МОДЕЛЬ ДАННЫХ

Универсальное хранилище 302 данных платформы 300 хранения, в соответствии с настоящим изобретением, реализует модель данных, которая поддерживает организацию, поиск, совместное использование, синхронизацию и безопасность данных, находящихся в хранилище. В модели данных, в соответствии с настоящим изобретением, фундаментальный (основной) модуль хранимой информации можно назвать элементом. Модель данных обеспечивает механизм декларирования элементов и расширений элемента и установления взаимосвязей между элементами, а также механизм организации элементов в папках и категориях.

В одном варианте выполнения настоящего изобретения модель данных основана на двух примитивных механизмах, Типы (Types) и Взаимозависимости (Relationships). Типы представляют собой структуры, которые обеспечивают формат, с помощью которого управляют формой экземпляра Типа. Формат выражен в виде упорядоченного набора Свойств (Properties). Свойство представляет собой имя значения или набора значений данного Типа. Например, тип USPostalAddress (Почтовый адрес США) может иметь свойства Street, City, Zip, State (улица, город, почтовый индекс, штат). Свойства могут быть необходимыми или дополнительными.

Взаимозависимости могут быть декларированы и могут представлять отображение между наборами примеров двух типов. Например, может быть декларирована Взаимозависимость между Типом Person (Человек) и Типом Location (Местоположение), называемая LivesAt (Проживает в), которая определяет, какие люди живут в каких местах. Взаимозависимость имеет имя, две конечных точки, а именно конечную точку - источник и конечную точку - цель (назначение).

Взаимозависимости могут также иметь упорядоченный набор свойств. И Конечная точка - источник, и Конечная точка - цель имеют Имя и Тип. Например, Взаимозависимость LivesAt имеет Источник (Source), называемый Occupant (Жилец), Типа Person и Цель, называемую Dwelling (Жилое помещение) Типа Location и, кроме того, имеет свойства StartDate (Дата начала) и EndDate (Дата окончания), указывающие период времени, в течение которого жилец проживает в этом жилом помещении. Следует отметить, что Person может жить во множестве жилых помещений в течение определенного времени, и жилое помещение может иметь множество

жильцов, так что наиболее вероятное место для размещения информации StartDate и EndDate представляет собой саму взаимосвязь.

Взаимозависимости определяют отображение между экземплярами, которые ограничены типами, заданными как типы конечной точки. Например, взаимозависимость LivesAt не может быть взаимозависимостью, в которой Automobile (Автомобиль) представляет собой Occupant, поскольку Automobile не является Person (Человеком).

1. ЭЛЕМЕНТЫ

Как указано выше, фундаментальный (основной) модуль сохраняемой информации в универсальном хранилище данных, в соответствии с настоящим изобретением, можно назвать элементом. Элемент представляет собой модуль сохраняемой информации, который в отличие от простого файла представляет собой объект, имеющий основной набор свойств, которые имеют общую поддержку для всех объектов, открытых (предоставленных) для конечного пользователя или прикладной программы платформой хранения. Для специалистов в данной области техники понятно, что универсальность универсального хранилища данных обеспечивается, частично благодаря тому, что каждый элемент в хранилище данных включает в себя данные, указывающие на эти основные свойства, сохраненные в соответствии со схемой данных, которая является постоянной для каждого элемента.

Универсальная схема данных обеспечивает универсальное основание, которое устанавливает концептуальные рамки для создания и организации элементов и свойств. Универсальная схема данных определяет некоторые специальные типы элементов и свойств, и признаки этих специальных фундаментальных (основных) типов, на основе которых могут быть дополнительно получены подтипы. Использование этой универсальной схемы данных позволяет программисту концептуально различать элементы (и их соответствующие типы) от свойств (и их соответствующих типов). Кроме того, универсальная схема данных определяет фундаментальный (основной) набор свойств, которыми могут обладать все элементы, поскольку все элементы (и их соответствующие Типы элементов) получены на основе этого фундаментального (основного) элемента в универсальной схеме данных (и их соответствующих Типов элементов). Благодаря сохранению каждого элемента в соответствии с данной универсальной схемой данных, программа просмотра оболочки позволяет интерпретировать и представлять для пользователя каждый элемент в хранилище данных, наряду с его основными свойствами. Пример универсальной схемы данных, пригодной для использования с настоящим изобретением, описан в находящейся в совместной собственности, одновременно находящейся на рассмотрении заявке "SYSTEM AND METHODS FOR REPRESENTING UNITS OF INFORMATION MANAGEABLE BY A HARDWARE/SOFTWARE INTERFACE SYSTEM BUT INDEPENDENT OF PHYSICAL REPRESENTATION", заявка на американский патент номер (еще назначен) (регистрационный номер патентного поверенного № MSFT-1748), подана 21 августа 2003 г., которая приведена здесь в качестве ссылки.

Элементы также имеют свойства и взаимозависимости (соотношения), для которых обеспечена общая поддержка для всех типов элементов, включая признаки, которые позволяют вводить новые свойства и взаимозависимости. Для специалистов в данной области понятно, что эти данные свойств и взаимозависимостей могут быть названы метаданными, ассоциированными с элементом. Как описано ниже, метаданные могут быть сохранены в соответствии со схемой художественного (декоративного)

оформления элемента. Эта схема художественного оформления элемента может указывать на соответствующий подход, используемый при представлении элемента для пользователя.

Элементы представляют собой объекты для общих операций, таких как копирование, удаление, перемещение, открытие, печать, сохранение резервной копии, восстановление, дублирование и т. д. Элементы представляют собой модули, которые могут быть сохранены и извлечены, и все формы сохраняемой информации, манипуляции с которой осуществляются платформой хранения, существуют в виде элементов, свойств элементов или взаимозависимостей между элементами, каждый из которых более подробно описан ниже.

Элементы предназначены для представления реальных слов и свободно понимаемых модулей данных, таких как Contacts, People, Services, Locations, Documents (Контакты, Люди, Услуги, Местоположения, Документы) (всех различных видов) и так далее.

Элементы представляют собой автономные объекты; таким образом, если удаляют один элемент, все свойства элемента также будут удалены. Аналогично, при извлечении элемента извлекают элемент и все его свойства, содержащиеся в метаданных элемента. В некоторых вариантах выполнения настоящего изобретения обеспечивается возможность запроса поднабора свойств при извлечении конкретного элемента; однако, по умолчанию для множества таких вариантов выполнения выдают элемент при его извлечении со всеми его непосредственными и унаследованными свойствами. Кроме того, свойства элементов также могут быть расширены путем добавления новых свойств к существующим свойствам этого типа элемента. Такие "расширения" ниже представляют собой истинные свойства элемента, и подтипы этого типа элемента могут автоматически включать в себя свойства расширения. Расширения также можно назвать метаданными, ассоциированными с файлом.

2. ПАПКИ И КАТЕГОРИИ ЭЛЕМЕНТА

Группы элементов могут быть организованы в специальные элементы, называемые Папками (Folders) элемента (которые не следует путать с папками файлов). В отличие от большинства файловых систем, однако, элемент может принадлежать более чем к одной Папке элемента, так что когда доступ к элементу выполняют и его обновляют в одной Папке элемента, доступ к этому обновленному элементу может быть выполнен затем непосредственно из другой папки элементов. По существу, хотя доступ к элементу может производиться из различных Папок элемента, то, к чему в действительности выполняется доступ, фактически представляет собой один и тот же элемент. Однако Папка элемента необязательно содержит все свои члены - элементы, или может попросту совместно содержать элементы вместе с другими папками, так что удаление папки элемента необязательно приводит к удалению элемента.

Элементы также могут принадлежать к Категориям на основе общих описанных характеристик, таких как (a) Тип (или Типы) элемента, (b) конкретное непосредственное или унаследованное свойство (или свойства), или (c) конкретное значение (или значения), соответствующее свойству элемента. Например, элемент, содержащий конкретные свойства для персональной контактной информации, может автоматически принадлежать к Категории Контакт, и любой элемент, содержащий свойства контактной информации, аналогично будет автоматически принадлежать к этой Категории. Аналогично, любой элемент, имеющий свойство местоположения со значением "New York City" (город Нью-Йорк) может автоматически принадлежать к Категории NewYorkCity.

Категории концептуально отличаются от Папок элементов тем, что в то время как папки элементов могут содержать элементы, которые не являются взаимосвязанными (то есть без общей описанной характеристики), каждый элемент в Категории имеет
 5 общий тип, свойство или значение ("общность"), которые описаны для этой Категории, и является такой общностью, которая формирует базис для его взаимозависимости с другими элементами в Категории и среди них. Кроме того, в то время как членство элемента в определенной Папке не является обязательным, в зависимости от любого конкретного аспекта этого элемента, для некоторых
 10 вариантов выполнения все элементы, имеющие общность, относящуюся к определенной Категории, могут автоматически стать членом этой Категории на уровне системы интерфейса аппаратных/программных средств. Концептуально категории также можно рассматривать как виртуальные Папки элементов, членство в которых основано на результатах конкретного запроса (как в контексте базы
 15 данных), и элементы, которые удовлетворяют условиям этого запроса (определенные по общности Категории), будут таким образом содержать членство в Категории.

На фиг.4 показана структурная взаимозависимость между элементами, Папками элементов и Категориями в различных вариантах выполнения настоящего
 20 изобретения. Множество элементов 402, 404, 406, 408, 410, 412, 414, 416, 418 и 420 являются членами различных Папок 422, 424, 426, 428 и 430 элементов. Некоторые элементы могут принадлежать более чем одной Папке элементов, например элемент 402 принадлежит Папкам 422 и 424 элементов. Некоторые элементы, например элемент 402, 404, 406, 408, 410 и 412 также являются членами одной или
 25 больше Категорий 432, 434 и 436, в то время как другие, например элементы 414, 416, 418 и 420, могут не принадлежать ни одной из Категорий (хотя это весьма маловероятно в некоторых вариантах выполнения, где наличие какого-либо свойства автоматически подразумевает членство в Категории, и, таким образом, у элемента
 30 должны полностью отсутствовать свойства для того, чтобы он не принадлежал ни к одной категории в таком варианте выполнения). В отличие от иерархической структуры папок, как Категории, так и Папки элементов имеют структуры, в большей степени похожие на ориентированные графы, как показано на чертеже. В любом случае все элементы, Папки элементов и Категории представляют собой элементы
 35 (хотя и с различными Типами элементов).

В отличие от файлов, папок и директорий, элементы, Папки элементов и Категории в соответствии с настоящим изобретением не имеют характерной "физической" природы, поскольку они не имеют концептуальных эквивалентов физических
 40 контейнеров, и, поэтому, элементы могут существовать в более чем одном таком местоположении. Способность элементов существовать в более чем одном элементе местоположения Папки, а также возможность организации по Категориям, обеспечивает улучшенную и расширенную степень манипулирования данными и
 45 дополнительные возможности в структуре хранения на уровне интерфейса аппаратных/программных средств, которые находятся за пределами возможностей, доступных в настоящее время в известном уровне техники.

3. ВЗАИМОЗАВИСИМОСТИ

Элементы также могут содержать реляционную (связанную) информацию, которая
 50 позволяет определять взаимозависимость между двумя или больше элементами. Взаимозависимости представляют собой бинарные взаимозависимости, где один элемент обозначается как источник и другой элемент обозначается как цель (назначение). Элемент - источник и элемент - цель взаимосвязаны с помощью

взаимозависимости. Взаимозависимости могут быть классифицированы как взаимозависимости Содержания и Ссылки. Взаимозависимости содержания управляют временем существования элементов - целей, в то время как взаимозависимости - ссылки не обеспечивают какой-либо семантики управления срока существования.

5 Типы взаимозависимостей Содержания дополнительно могут быть разделены на взаимозависимости Удержания и Встраивания. Взаимозависимость Удержания управляет сроком существования цели с использованием механизма учета ссылки. Взаимозависимости Удержания не содержат свои цели, но управляют сроками
10 существования целей. Когда все взаимозависимости удержания с элементом удаляют, этот элемент также удаляют. Взаимозависимость Встраивания позволяет моделировать составные элементы, и ее можно рассматривать как исключительные взаимозависимости удержания. Элемент может представлять собой цель (назначение) для одной или больше взаимозависимостей удержания, но элемент может
15 представлять собой цель только одной взаимозависимости встраивания. Элемент, который представляет собой цель для взаимозависимости встраивания, не может быть целью любой другой взаимозависимости удержания или встраивания. Встроенные взаимозависимости содержат свои цели и управляют сроком существования целей. Для
20 специалистов в данной области понятно, что одиночная цель может быть, в самое большее, одной встроенной взаимозависимости, в то время как одиночная цель может быть во множестве взаимозависимостей удержания.

Взаимозависимости Ссылки не управляют сроком существования элемента - цели. Они могут быть свободными ("висящими"), элемент-цель может не существовать.
25 Взаимозависимости Ссылки можно использовать для моделирования ссылок на элементы в любом месте в глобальном пространстве имен элементов (то есть, включая удаленное хранилище данных).

Выборка элемента не приводит автоматически к выборке его взаимозависимостей.
30 Прикладные программы или оболочка должны явно запрашивать взаимозависимость элемента. Кроме того, модификация взаимозависимости не приводит к модификации элемента - источника или цели; аналогично, добавление взаимозависимости не влияет на элемент - источник/цель. Взаимозависимости между двумя элементами могут быть декларированы и сохранены с элементом или оболочкой, или с прикладной
35 программой, хотя путем использования реляционной информации можно определить, что эти два элемента являются взаимосвязанными.

Взаимозависимость Ссылки не обеспечивает управление временем существования элемента, на который она ссылается. Более того, взаимозависимости ссылки не
40 гарантируют существования цели, и при этом они также не гарантируют тип цели, указанной при декларации взаимозависимости. Это означает, что взаимозависимости Ссылки могут быть свободными ("висящими"). Кроме того, взаимозависимость ссылки может ссылаться на элементы, находящиеся в других хранилищах данных. Взаимозависимости ссылки можно рассматривать как концепцию, аналогичную
45 ссылкам (связям) в web - страницах.

В по меньшей мере одном варианте выполнения платформа хранения в соответствии с настоящим изобретением поддерживает упорядочение взаимозависимостей. Упорядочение обеспечивается свойством, названным "Порядок"
50 (Order). В поле Порядок отсутствует однозначность ограничения. Порядок взаимозависимостей с одинаковым значением свойства "порядок" не является гарантированным, однако гарантируется, что их можно упорядочить после взаимозависимостей с более низким значением поля "порядок" и перед

взаимозависимостями с более высоким значением поля "порядок". Следует отметить, что свойство "Порядок" не является базовым определением взаимозависимости. Скорее оно является внешним свойством, которое сохранено как часть взаимозависимости между источником и целью.

Как указано выше, элемент может представлять собой член Папки элемента. Что касается Взаимозависимостей, элемент может иметь взаимозависимость с Папкой элементов. В нескольких вариантах выполнения настоящего изобретения некоторые взаимозависимости представлены Взаимозависимостями, существующими между элементами.

4. ВОЗМОЖНОСТЬ РАСШИРЕНИЯ

Как показано на фиг.3, платформа хранения содержит исходный набор схем 340, как описано выше. Однако, кроме того, по меньшей мере в некоторых вариантах выполнения платформа хранения позволяет клиентам, включая независимого поставщика программных средств (НПП, ISV), создавать новые схемы 344.

С. МЕХАНИЗМ БАЗЫ ДАННЫХ

Как указано выше, хранилище данных выполнено на основе механизма базы данных. В одном варианте выполнения механизм базы данных содержит механизм реляционной базы данных, в котором используется язык запросов SQL (ЯСЗ, язык структурированных запросов), такой как Microsoft SQL Server engine, с объектными реляционными расширениями. Однако следует понимать, что можно использовать другие механизмы базы данных. Действительно, помимо выполнения модели концептуальных данных платформы хранения на основе механизма реляционной базы данных, она также может быть выполнена на основе баз данных других типов, например на основе объектно-ориентированной базы данных и баз данных XML (расширяемый язык разметки гипертекста).

III. ПРЕДСТАВЛЕНИЕ ЭЛЕМЕНТОВ ДЛЯ ПОЛЬЗОВАТЕЛЯ

Элементы универсального хранилища данных представляют для пользователя с помощью программы просмотра (браузера) оболочки. Такие программы просмотра хорошо известны в данной области техники и, как поясняется выше, программа просмотра оболочки обеспечивает интерфейс пользователя, позволяющий пользователю просматривать аппаратно/программный интерфейс и взаимодействовать с ним.

А. ВИД ОТОБРАЖЕНИЯ ПО УМОЛЧАНИЮ

Как указано выше, каждый элемент в универсальном хранилище данных сохранен в соответствии с универсальной схемой данных. Эта схема включает в себя механизм для описания элементов, называемый ассоциациями типа. Каждая ассоциация типа имеет базовое представление в оболочке; благодаря сохранению элемента, в соответствии с ассоциацией типа, оболочка позволяет отображать элемент, в соответствии с, по меньшей мере, базовым видом отображения или с видом отображения по умолчанию.

Ассоциация типа представляет собой свойство, ассоциированное с элементом; при помещении данных в универсальное хранилище данных одно или больше свойств, ассоциированных с данными, должно быть декларировано для определения типа элемента. Эти свойства могут быть включены как метаданные, ассоциированные с данными. Оболочка содержит набор принятых по умолчанию ассоциаций типа, которые представляют самые основные (базовые) и минимальные свойства, которые должны быть декларированы для элемента.

На фиг.5 показан элемент 500. Элемент 500 записан в соответствии с универсальной

схемой данных и включает в себя набор данных 502 элемента и набор метаданных 504, включая декларацию свойств. Данные 502 элемента могут представлять собой любой набор данных, подходящих для включения в хранилище данных. Например, данные 502 элемента могут быть ассоциированы с редактируемым текстовым процессором документом. Метаданные 504 декларации свойства включают в себя, по меньшей мере, декларацию основного (базового) типа для элемента 500. Например, ассоциация принятого по умолчанию типа может представлять собой тип Документ (Document), и метаданные 504 могут устанавливать, что элемент 500 является элементом типа Документ. Поскольку оболочка включает в себя вид отображения, принятый по умолчанию, для каждой ассоциации принятого по умолчанию типа оболочка может отображать элемент 500 в соответствии с принятым по умолчанию видом отображения для типа Документ. Принятый по умолчанию вид отображения для типов Документы может, например, включать в себя иконку (пиктограмму), используемую только с элементами типа Документ. Благодаря представлению элемента редактируемого текста с помощью такого изображения, пользователь может быстро распознавать, что элемент 500 представляет собой документ. Для специалистов в данной области техники понятно, что любые разновидности ассоциаций принятого по умолчанию типа и атрибутов отображения принятого по умолчанию элемента могут быть приемлемыми для настоящего изобретения.

В. ВИД ХУДОЖЕСТВЕННОГО ОФОРМЛЕНИЯ ЭЛЕМЕНТА

Помимо деклараций свойства, метаданные, ассоциированные с элементом, могут включать в себя данные, указывающие на художественное (декоративное) оформление представления элемента, обеспечиваемое оболочкой. В качестве художественного оформления в данном случае можно считать "указания" (подсказки), направленные на представление элемента для пользователя. Такие метаданные могут быть сохранены в соответствии со схемой художественного оформления элемента. Схема художественного оформления элемента определяет вид художественного оформления элемента, который оболочка может использовать для представления элемента. Например, данные художественного оформления элемента могут описывать наиболее важные декларируемые свойства элемента. Эти свойства "высокой значимости" могут быть наиболее желательными для представления в оболочке.

Элемент 500, в случае необходимости, может включать в себя данные 506 художественного оформления, сохраненные в соответствии со схемой художественного оформления элемента. Для представления элемента 500, данные 506 художественного оформления элемента могут указывать на набор полей просмотра, соответствующих представлению элемента 500. Поля просмотра являются проекцией декларируемых свойств, и общие поля просмотра могут включать в себя "заголовок", "автор", "дата создания" или "дата последнего редактирования". Оболочка включает в себя набор стандартных полей просмотра, и независимые производители программных средств (ISV) могут определять поля просмотра, которые являются подходящими представлению их данных. При разработке новых типов элементов ISV могут либо отображать свойства элемента, которые они определяют в полях просмотра оболочки, или они могут создавать свои собственные поля просмотра.

Например, данные 502 элемента могут содержать данные песни. Набор декларируемых свойств 506 может включать в себя свойства, такие как название песни, имя артиста, дату записи, альбом, длительность песни и другие декларации,

соответствующие такому элементу песни. Данные 506 художественного оформления элемента могут указывать, что поля просмотра "Название", "Артист" и "Альбом" должны отображаться для пользователя при представлении элемента 500 в оболочке.

Данные 506 художественного оформления элемента могут описывать элементы, имеющие в большей степени истинную декоративную функцию в отношении данных 502 элемента, таких как текст, представленный с декларируемым свойством. Например, одна из деклараций 504 свойства может указывать значение скорости передачи битов для описания качества записи. Это свойство может быть записано как целое число BITRATE (скорость передачи битов). Данные 506 художественного оформления элемента могут запрашивать значение скорости передачи битов для отображения и также могут декорировать это поле в виде "[BITRATE] килобайт в секунду". В этом способе поле скорости передачи имеет соответствующее художественное оформление, что позволяет пользователю легко воспринимать значение величины скорости передачи битов в поле просмотра.

Для специалистов в данной области техники понятно, что данные 506 художественного оформления элемента и их соответствующий вид художественного оформления элемента могут определять широкое разнообразие атрибутов представления. Художественные оформления элемента могут составлять любой аспект отображения, поддерживаемый оболочкой. Некоторые общие другие виды художественного оформления элемента представляют собой, например, форматирование данных, заданный по умолчанию порядок сортировки и заданный по умолчанию размер изображения. Кроме того, данные 506 художественного оформления элемента могут описывать общие элементы управления, используемые при отображении данного элемента. Например, в поле Ratings (рейтинги) можно использовать элементы управления рейтинга, которые представляют рейтинг в виде последовательности звездочек. Данные 506 художественного оформления элемента могут описывать задачи или команды, соответствующие использованию этого элемента. Для специалистов в данной области техники понятно, что термины "задача" и "команда" описывают некоторое действие, предпринимаемое в отношении элемента и такие термины можно использовать взаимозаменяемо. Например, команды "Редактировать" или "Предварительный просмотр" могут использоваться в качестве соответствующих задач/команд, ассоциированных с элементом. Оболочка может быть дополнительно сконфигурирована так, что она будет запускать прикладные программы, поддерживающие эти задачи при выборе пользователем действия, выполняемого в отношении к элементу.

Для специалистов в данной области техники понятно, что художественное оформление элемента будет изменяться и расти с течением времени. В настоящем изобретении предусматривается, что когда разрабатывают новые художественные оформления элемента, эти новые художественные оформления можно использовать для новых элементов, в то время как для старых элементов отображение свойств будет продолжаться с использованием старых атрибутов отображения, обеспечиваемых оболочкой.

На фиг.6 показана схема последовательности выполнения операций, иллюстрирующая способ 600 представления элементов для пользователя в соответствии с настоящим изобретением. На этапе 602 согласно способу 600 выполняют доступ к универсальному хранилищу данных в соответствии с запросом на представление одного или больше элементов для пользователя. На этапе 604 способа выполняется анализ одного или больше элементов, выбранных для

представления. Элементы, содержащие метаданные, сохраненные в соответствии со схемой художественного оформления элемента, представляют пользователю в соответствии с видом художественного оформления элемента, как обозначено на этапе 606. Элементы, которые не содержат такие метаданные, представляют в соответствии с определенным по умолчанию видом отображения, как указано на этапе 608. Для специалистов в данной области техники понятно, что, как указано выше, эти схемы и виды представления могут включать в себя различные атрибуты элемента отображения, которые можно использовать для представления каждого элемента в хранилище данных.

С. ВИД ОБОЛОЧКИ

Как описано выше, вид художественного оформления элемента является достаточным для полного представления данного элемента или однородного набора элементов, состоящего из элементов, имеющих одинаковый вид художественного оформления элемента. Для отображения элементов с различными схемами художественного оформления элемента оболочка обеспечивает схемы вида оболочки, которые представляют элементы в соответствии с видами художественного оформления оболочки. Схема вида оболочки позволяет оболочке или ISV декларировать соответствующие виды для заданных наборов однородных данных.

Элементы, выбранные для представления в виде художественного оформления оболочки, могут включать в себя общую характеристику. Для специалистов в данной области техники понятно, что широкое разнообразие общих характеристик может быть приемлемым для вида художественного оформления оболочки. Например, схема вида оболочки может определять вид "Изображение", используемый для отображения общих и соответствующих полей и метаданных для всех известных типов изображений (например, GIF, .JPEG, .BMP, .TIFF и т.д.). Схема вида оболочки доминирует над конфликтующими атрибутами элемента изображения для заданного вида художественного оформления элемента и представляет каждый элемент изображения в соответствии со схемой вида оболочки. В качестве другого примера, оболочка может обеспечить вид оболочки "Документ", оптимизированный по соответствующим колонкам, и метаданные для элементов, формируемых при использовании обычных офисных приложений, таких как документы текстовых редакторов, электронных таблиц или баз данных, даже если варианты художественного оформления каждого из этих элементов могут существенно отличаться друг от друга. Такой вид имеет значение в соответствии с общими свойствами каждого из этих документов. Для специалистов в данной области техники понятно, что при установлении последних типов документов вид оболочки позволяет представлять эти новые элементы в соответствии с согласованным видом оболочки, даже если новый тип не был учтен при первоначальном создании этого вида.

В дополнение к формированию полей просмотра, подходящих для заданного набора однородных элементов, схема вида оболочки может определять дополнительные атрибуты элемента отображения. Например, состояние представления, включая свойства пиктограмм, размер ячейки предварительного просмотра и принятый по умолчанию порядок сортировки могут быть определены схемой вида оболочки. Схема вида оболочки также содержит элементы художественного оформления свойств, такие как форматирование данных, которые применяют к различным колонкам.

В случаях, когда схема вида оболочки и схема художественного оформления элемента находятся в конфликте, схема вида оболочки преобладает (доминирует). В

случаях, когда элемент отображения отсутствует в схеме вида оболочки, вид оболочки переходит обратно к схеме художественного оформления элемента для соответствующего отображения. Таким образом, вид оболочки позволяет
 5 выстраивать соответствующий вид при отображении данных, не предусмотренных первоначально. Кроме того, в одном варианте выполнения настоящего изобретения вид оболочки может подчиняться виду художественного оформления элемента для получения не находящегося в конфликте элемента художественного оформления. Например, вид оболочки может использовать метаданные с "высокой значимостью"
 10 из набора элементов, которые их содержат, для построения соответствующего набора колонок и метаданных для отображения элементов.

Для специалистов в данной области техники понятно, что схемы вида оболочки могут обеспечивать широкое разнообразие атрибутов отображения и что ISV могут захотеть обеспечить такие виды оболочки. Атрибуты отображения могут включать в
 15 себя, без ограничения: размер плана (сцены) предварительного просмотра, метаданные, предназначенные для отображения в плане предварительного просмотра, используемые специальные элементы управления, задачи и команды, соответствующие представленным элементам.

На фиг.7 показана схема последовательности выполнения операций, иллюстрирующая способ 700 представления для пользователя элементов в соответствии с настоящим изобретением. На этапе 702 в способе 700 выполняется доступ к хранилищу данных в соответствии с запросом на представление для
 20 пользователя одного или больше элементов. На этапе 704 выполняется определение, содержит ли каждый элемент, выбранный для представления, одинаковую схему художественного оформления элемента. Если все элементы имеют такую общую
 25 схему, эти элементы представляют в соответствии с этой схемой, как обозначено на этапе 706. Если представляемые элементы содержат элементы, имеющие различные схемы отображения или не содержат схемы отображения, на этапе 708 набор
 30 элементов представляют в соответствии с видом художественного оформления оболочки. Как описано выше, такой вид оболочки может быть подходящим для представления однородных наборов элементов. Необязательный этап представления одного или больше выбранных элементов с отображением элементов из вида
 35 художественного оформления элемента включен на этапе 710. Хотя эти элементы отображения могут не находиться в конфликте с видом художественного оформления оболочки, такие элементы могут улучшать представление элементов оболочки.

На фиг.8 показана схема примера иерархии 800 схемы вида. Нижний уровень иерархии представляет собой схему 802 вида элемента. Схема 802 вида элемента
 40 обеспечивает базовое отображение, требуемое для представления элемента или, если схема вида не обеспечена, обеспечивает отображение по умолчанию. Схемы, которые расположены над схемой 802 вида элемента, когда это требуется, могут подчиняться или прибегать к своим элементам отображения.

Схема 804 вида оболочки расположена над схемой 802 вида элемента. Как описано выше, схему вида оболочки можно использовать для отображения набора элементов с
 45 различными видами элемента. Схема вида оболочки определяет задачи 806, характеристики 808 плана предварительного просмотра, колонки 810 и художественное оформление 812, которые используются для отображения элементов в
 50 соответствии с видом 804 оболочки. Вид оболочки может прибегать к схеме 802 вида элемента для обеспечения не находящихся в конфликте художественных оформлений элемента для использования с видом 804 оболочки. Кроме того, установки 814 вида

пользователя могут размещаться в виде оболочки. Эти установки представляют требуемый для пользователя формат представления элементов оболочки. Для специалистов в данной области техники понятно, что в схеме отображения может быть определено любое количество атрибутов отображения, и что для пользователя могут быть представлены различные варианты выбора и элементы управления установок отображения.

D. ВИД ОТОБРАЖЕНИЯ ПРОВОДНИКА

Оболочка также может быть сконфигурирована для представления элементов в соответствии с видом отображения проводника. "Проводником" может называться прикладная программа хранения и он может предоставляться оболочкой или ISV. В одном варианте выполнения настоящего изобретения проводник может быть создан для создания целостного опыта, который помогает пользователям управлять большим набором элементов. Например, проводник может обеспечить пользователю возможность просмотра, запроса, навигации, включения в задачи или организации выбранных элементов в хранилище данных. Термин "проводник" не должен подразумевать местоположение, где расположены отображаемые элементы, и такие термины как "центр активности", "средство просмотра" и "библиотека" можно использовать взаимозаменяемо с термином "проводник" для описания прикладной программы хранения в соответствии с настоящим изобретением.

На фиг.9 показан пример иерархии 900 схемы проводника. Нижний уровень иерархии представляет собой схему 902 вида элемента. Схема 902 вида элемента обеспечивает базовое отображение, требуемое для представления элемента, и схема 904 вида проводника может подчиняться или прибегать к своим элементам отображения, когда это требуется.

Схема вида проводника включает в себя схему 906 вида оболочки и художественное оформление 908 проводника. Художественное оформление 908 проводника декорирует проводник в целом и создает элементы отображения, такие как отличительные цвета и элементы фирменного обозначения. Это художественное оформление 908 проводника продолжает существовать среди различных видов, которые обеспечивает проводник. Для специалистов в данной области техники понятно, что широкое разнообразие атрибутов отображения можно использовать для художественного оформления 908 проводника. Например, запросы данных или задачи/команды, ассоциированные с элементами проводника, можно использовать для отображения с помощью проводника. Отображаемые задачи, предпочтительно, должны быть связаны с прикладными программами, которые позволяют выполнять эти задачи.

Схема вида проводника, необязательно, может включать в себя схему 906 вида оболочки или множество схем вида оболочки. Схема 906 вида оболочки может быть сконфигурирована для предоставления вида оболочки для поднабора элементов проводника. Например, проводник может быть сконфигурирован для отображения для пользователя отдельных песен. Первая схема вида оболочки может быть включена для обеспечения отображения альбомов, и вторая схема вида оболочки может быть включена для обеспечения отображения дорожек песни. Таким образом, оба типа элементов будут иметь соответствующие виды в проводнике. Как описано выше, использование вида оболочки относится к представлению наборов элементов, которые необязательно могут иметь общую характеристику.

Проводник также может быть основан на видах оболочки, включенных в оболочку. Если элементы, выбранные для представления в проводнике, не

поддерживаются ни одним из видов оболочки, включенных в проводник, оболочка может обеспечить соответствующий вид оболочки для использования в проводнике. Аналогично и как описано выше, проводник также может прибегать к виду отображения элемента или заданному по умолчанию виду отображения, обеспечиваемому оболочкой. Эта функция обеспечивает то, что любой элемент, который может быть отображен оболочкой, также может быть отображен в проводнике. Проводник может быть сконфигурирован подчиняться схемам отображения, предоставляемых оболочкой, или может быть основан на них, для обеспечения, например, отображения непредвиденных данных.

На фиг.10 представлен способ 1000 представления элементов при отображении проводника в соответствии с настоящим изобретением. На этапе 1002 в способе 1000 выполняют доступ к хранилищу данных, и на этапе 1004 выбирают элементы для отображения в проводнике. Выбор элементов проводника может быть основан на анализе деклараций элемента, и также называется записями поля. Как описано выше, элементы в хранилище данных могут содержать информацию о свойствах. Эту информацию декларируют, когда элемент размещают в хранилище данных, и ее можно обновлять в течение срока существования элемента. Такие декларации можно рассматривать, как записи поля, соответствующие набору полей свойства. Например, записи в поле свойства "автор" могут содержать информацию об авторе данного элемента.

Может быть необходимо представлять элементы, совместно использующие одну или больше общих записей поля. Например, может потребоваться проводник, включающий каждый элемент, автором которого является конкретный человек. Благодаря рассмотрению записи в поле автора, проводник позволяет выбирать из хранилища данных такие элементы проводника, автором которых является конкретный человек. Для специалиста в данной области техники хорошо известна технология выполнения такого запроса в базе данных.

На этапе 1006 выполняется определение, включает ли проводник вид оболочки, который является подходящим для представления элемента проводника. Если такой подходящий вид оболочки не будет найден в проводнике, в способе 1000 используется вид, содержащийся в оболочке, как обозначено на этапе 1008. Если нужный вид оболочки включен в проводник, в способе 1000 на этапе 1010 используется этот вид оболочки для представления элемента. На этапе 1012 выполняется дополнительный этап использования декоративных элементов из схемы отображения элемента. Как описано выше, в проводнике можно использовать не конфликтующие художественные оформления из схемы вида элемента для улучшения представления элемента. На этапе 1014 элемент проводника представляют пользователю в соответствии с видами оболочки и видами элементов. Декоративные свойства проводника представлены на этапе 1016. Эти свойства для отображения атрибутов могут быть самыми разнообразными и могут включать запросы данных или задачи, ассоциированные с элементами проводника.

Е. УСОВЕРШЕНСТВОВАНИЕ ПРОВОДНИКА

Проводники могут быть созданы для широкого разнообразия типов элементов. В одном варианте выполнения настоящего изобретения проводники могут быть определены с использованием незначительного объема программирования или без программирования. Благодаря обеспечению возможности создания проводников в зависимости от данных упрощается разработка при обеспечении согласованного внешнего вида и восприятия проводников.

В некоторых вариантах выполнения проводники могут накладывать ограничения в отношении типов (включая расширения элементов и расширения файлов) элементов, которые они могут представлять или проводники могут выбирать для разрешения элементов всех типов. Кроме того, в проводниках может быть установлена
 5 возможность разрешения типов элементов с конкретным набором расширений элемента. Например, проводник элемента Закон может отображать все элементы с прикрепленным свойством "LegalItemExtension". В проводниках может быть включено разрешение элементов некоторого типа и с любым расширением файла, которое
 10 отображается для этого типа. Например, проводник музыкальных элементов может отображать расширения всех музыкальных файлов, таких как mp3 или wma. Кроме того, в проводниках может быть включено разрешение элементов только с некоторым набором расширений файла. Если проводник ограничен определенным набором типов, элементы других типов не могут быть сохранены или перемещены в
 15 этот проводник. Проводники могут повторно определять ассоциации типов для типов, которые в них разрешены, в проводниках может быть запрещено осуществлять выборочное доминирование, и в проводниках может быть включен запрет добавления новых команд. Кроме того, проводники могут определять, позволяют ли они
 20 конечным пользователям преобладать над ассоциациями типа в проводнике.

С учетом вышеприведенного, для специалистов в данной области техники понятно, что благодаря созданию методики разработки в зависимости от данных для создания проводника, предназначенного для использования в среде, основанной на типе
 25 элемента, проводники могут быть определены декларативно и без использования специальной программы.

На фиг.11 представлен способ 1100 представления элементов в соответствии со схемой отображения проводника. На этапе 1102, в способе 1100 выполняется выбор
 30 требуемой записи поля. Как описано выше, такая требуемая запись поля может соответствовать декларированному свойству, ассоциированному с элементом. Например, "проводник" фотоальбома может иметь требуемую запись поля, в которую требуется включать элементы, содержащие данные изображения.

На этапе 1104 определяется схема отображения проводника. Эта схема
 35 отображения может включать в себя схему вида оболочки и художественное оформление проводника. Художественное оформление проводника украшает проводник в целом и обеспечивает элементы отображения, такие как отличающие цвета и элементы фирменного обозначения. Эти художественные оформления
 40 проводника сохраняются среди различных видов, которые обеспечивает проводник. Широкое разнообразие атрибутов отображения можно использовать для художественного оформления проводника. Например, запросы данных или задачи/команды, ассоциированные с элементами проводника, можно использовать
 45 для отображения с помощью проводника. Отображаемые задачи, предпочтительно, будут связаны с приложением, которое позволяет выполнять эти задачи. Схема вида проводника может включать в себя схему вида оболочки или множество схем вида оболочки. Схема просмотра вида может быть сконфигурирована для обеспечения
 вида оболочки для поднабора элементов проводника.

На этапе 1106, в способе 1100 выполняется доступ к хранилищу данных для выбора
 50 элементов проводника. Элементы проводника ассоциированы с требуемой записью поля. Для специалистов в данной области техники понятно, что выбор таких элементов в базе данных хорошо известен в данной области техники. Для специалистов в данной области техники также понятно, что разработчики могут

создать проводники, сконфигурированные для такого взаимодействия. Разработка проводников, позволяющих выполнять доступ к хранилищу данных, рассматривается в настоящем изобретении.

На этапе 1108 элементы проводника отображаются в соответствии со схемой отображения проводника. Схема отображения проводника описана выше, и такое отображение также может включать в себя взаимодействие с программой вида оболочки. Например, оболочка может обеспечивать один или больше видов оболочки. Кроме того, элементы художественного оформления элемента из схемы художественного оформления элемента могут использоваться для расширения представления элементов проводника.

Для специалистов в данной области техники понятно, что прикладная программа хранения может рассматриваться как прикладная программа и/или расширение программы просмотра оболочки. Следовательно, вышеприведенное описание является подходящим для обоих представлений настоящего изобретения. В качестве программы приложения программа проводника может включать в себя модуль взаимодействия с оболочкой, который сконфигурирован для взаимодействия с программой просмотра из оболочки. Такое взаимодействие позволяет программе передавать информацию в оболочку и позволяет программному обеспечению работать вместе с представленными элементами. Модуль взаимодействия оболочки может облегчать доступ к хранилищу данных и может предоставлять атрибуты отображения. Такое взаимодействие между прикладной программой и оболочкой хорошо известно в данной области техники.

Ф. ПРЕДСТАВЛЕНИЕ ВЗАИМОСВЯЗАННЫХ ЭЛЕМЕНТОВ

Настоящее изобретение также позволяет отображать для пользователя взаимосвязанные элементы в хранилище данных. Как описано выше, элементы в хранилище данных могут включать в себя элементы, имеющие одно или больше декларированных свойств. Элемент может иметь декларированные взаимозависимости, которые поясняют другие элементы в хранилище данных, имеющих общую взаимозависимость. Например, элемент, содержащий адрес электронной почты, может декларировать взаимосвязь с элементом, содержащим другую контактную информацию, относящуюся к владельцу адреса электронной почты. В оболочке может использоваться эта декларированная взаимосвязь для представления другой контактной информации по запросу пользователя. Оболочка также может определять взаимозависимости путем рассмотрения декларированных свойств элемента. Например, набор документов может быть взаимозависимым, если документы имеют общие свойства; элементы с расширением "LegalItemExtension" могут быть взаимозависимыми, если общее значение сохранено как часть расширения. Такая взаимозависимость может быть определена по запросу данных, как хорошо известно в данной области техники.

На фиг.12 представлен способ 1200, предназначенный для представления взаимозависимых элементов в соответствии с настоящим изобретением. На этапе 1202, в способе 1200 выполняется доступ к хранилищу данных и на этапе 1204 определяется взаимозависимость между элементами в хранилище данных. Как описано выше, при таком определении используют декларированные свойства, включенные в элемент. Такое определение может быть выполнено в ответ на команду пользователя. Например, для пользователя может отображаться элемент, имеющий набор декларированных характеристик элемента. Характеристическая и реляционная (связанная) информация элемента может отображаться вместе с элементом.

Пользователь может выбрать одну из характеристик и может вводить запрос на просмотр других элементов, имеющих общую характеристику элемента. На этапе 1206, в способе 1200 для пользователя представляют взаимосвязанные элементы. Такое представление может включать любую схему отображения, известную в данной области техники.

Для специалистов в данной области техники будут очевидны альтернативные варианты выполнения и воплощения настоящего изобретения, к которому оно относится, после чтения описания совместно с рассмотрением чертежей. В соответствии с этим объем настоящего изобретения определяется прилагаемой формулой изобретения, а не приведенным выше описанием.

Формула изобретения

1. Компьютерная система для представления пользователю взаимосвязанных элементов в запоминающем устройстве данных, содержащая:

запоминающее устройство данных, содержащее множество элементов, сохраненных в соответствии со схемой данных, определяющей множество элементов и взаимозависимости между элементами, и содержащих реляционную информацию, соответствующую по меньшей мере части упомянутого множества элементов, причем упомянутая реляционная информация позволяет определять взаимозависимости между двумя или более из множества элементов,

при этом упомянутое запоминающее устройство данных использует упомянутую реляционную информацию для удаления одного или более элементов из упомянутого запоминающего устройства данных в ответ на изменение в по меньшей мере части упомянутых взаимозависимостей; и

оболочку для представления упомянутого множества элементов пользователю, при этом оболочка выполнена с возможностью представления выбранного элемента для пользователя и дополнительно выполнена с возможностью использования упомянутой реляционной информации для представления одного или более элементов, которые взаимосвязаны с упомянутым выбранным элементом, в упомянутом запоминающем устройстве данных.

2. Компьютерная система по п.1, в которой реляционная информация, соответствующая одному или более из упомянутого множества элементов, включает в себя набор характеристик элемента.

3. Компьютерная система по п.2, в которой упомянутая оболочка сконфигурирована для представления пользователю одной или более характеристик из упомянутого набора характеристик элемента.

4. Компьютерная система по п.2, в которой упомянутая оболочка выполнена с возможностью приема команд пользователя, представляющих выбор для просмотра одного или более элементов, в запоминающее устройство данных, имеющих одну из упомянутых характеристик элемента.

5. Компьютерная система по п.2, в которой упомянутая оболочка выполнена с возможностью представления одного или более элементов в запоминающем устройстве данных, которые имеют общую одну из упомянутых характеристик элемента.

6. Компьютерная система по п.1, в которой оболочка выполнена с возможностью представления по меньшей мере части упомянутой реляционной информации.

7. Компьютерная система по п.1, в которой оболочка выполнена с возможностью приема входных данных пользователя, представляющих выбор для просмотра

элементов в запоминающем устройстве данных, которые взаимосвязаны с упомянутым выбранным элементом.

8. Компьютерная система по п.7, в которой упомянутая реляционная информация, соответствующая выбранному элементу, включает в себя набор характеристик элемента, связанных с выбранным элементом, и в которой упомянутые входные данные пользователя представляют выбор для просмотра одного или более элементов в запоминающем устройстве данных, которые имеют одну общую из упомянутого набора характеристик элементов с выбранным элементом.

9. Компьютерная система по п.2, в котором оболочка содержит: компонент взаимодействия с упомянутым запоминающим устройством данных, который извлекает данные, связанные с упомянутыми одним или более элементами, из упомянутого запоминающего устройства данных, причем по меньшей мере часть упомянутых взаимозависимостей между двумя или более элементами имеют ассоциированные семантики управления временем существования,

компонент представления пользователю взаимосвязанного элемента, который использует упомянутые извлеченные данные для представления пользователю упомянутых взаимосвязанных элементов, причем компонент представления взаимозависимости выполнен с возможностью представлять выбранный элемент пользователю и дополнительно выполнен с возможностью использовать упомянутую реляционную информацию для представления одного или более элементов в упомянутом запоминающем устройстве данных, которые взаимосвязаны с упомянутым выбранным элементом, и

средство управления временем существования элемента, которое использует упомянутые ассоциированные семантики управления временем существования для удаления одного или более элементов из упомянутого запоминающего устройства данных в ответ на изменение в по меньшей мере части упомянутых взаимозависимостях.

10. Компьютерная система по п.9, в которой упомянутый компонент представления взаимосвязанного элемента выполнен с возможностью представления пользователю одной или более из упомянутого набора характеристик элемента.

11. Компьютерная система по п.9, в которой упомянутый компонент представления взаимосвязанного элемента выполнен с возможностью представления пользователю одного или более элементов в запоминающем устройстве данных, которые имеют общую одну из упомянутых характеристик элемента.

12. Компьютерная система по п.9, в которой упомянутый компонент представления взаимосвязанного элемента выполнен с возможностью представления по меньшей мере части упомянутой реляционной информации.

13. Реализуемый на компьютере способ представления для пользователя взаимосвязанных элементов в запоминающем устройстве данных, содержащий этапы: осуществляют доступ к данным в упомянутом запоминающем устройстве данных, причем упомянутое запоминающее устройство данных хранит множество элементов в соответствии со схемой данных, определяющей упомянутое множество элементов и взаимозависимости между элементами, и причем по меньшей мере часть упомянутого множества элементов содержит реляционную информацию, которая позволяет определять взаимозависимости между упомянутым множеством элементов;

используют упомянутую реляционную информацию для определения взаимозависимости между выбранным элементом и одним или более элементами, содержащими упомянутую реляционную информацию, в запоминающем устройстве

данных;

отображают пользователю упомянутый выбранный элемент и один или более взаимосвязанных элементов,

принимают пользовательские входные данные, вызывающие изменения в упомянутой взаимозависимости, и

удаляют по меньшей мере один из упомянутого множества элементов из упомянутого запоминающего устройства данных в ответ на упомянутое изменение.

14. Способ по п.13, в котором отображение упомянутого выбранного элемента и одного или более взаимосвязанных элементов включает в себя отображение для пользователя по меньшей мере части упомянутой реляционной информации.

15. Способ по п.13, в котором упомянутый способ дополнительно содержит прием входных данных пользователя, представляющих выбор для просмотра одного или более элементов, в запоминающем устройстве данных, которые взаимосвязаны с упомянутым выбранным элементом.

16. Способ по п.15, в котором отображение упомянутого выбранного элемента и одного или более взаимосвязанных элементов выполняют в ответ на упомянутые входные данные.

17. Считываемый компьютером носитель, имеющий выполняемые компьютером инструкции для выполнения способа для представления пользователю взаимосвязанных элементов в запоминающем устройстве данных, содержащего этапы:

осуществляют доступ к данным в упомянутом запоминающем устройстве данных, причем упомянутое запоминающее устройство данных хранит множество элементов в соответствии с универсальной схемой данных, определяющей упомянутое множество элементов и взаимозависимости между элементами, и причем по меньшей мере часть упомянутого множества элементов содержат реляционную информацию, которая позволяет определить взаимозависимости между двумя или более из упомянутого множества элементов, при этом по меньшей мере часть упомянутых взаимозависимостей обозначают один или более исходных элементов и один или более целевых элементов;

используют упомянутую реляционную информацию для определения взаимозависимости между выбранным элементом и одним или более элементами, содержащими упомянутую реляционную информацию, в запоминающем устройстве данных;

представляют пользователю упомянутый выбранный элемент и один или более взаимосвязанных элементов,

принимают пользовательские входные данные, изменяющие по меньшей мере один из упомянутых одного или более исходных элементов или изменяют по меньшей мере часть упомянутых взаимозависимостей, и

удаляют по меньшей мере один из упомянутых одного или более целевых элементов из упомянутого запоминающего устройства данных, если упомянутый по меньшей мере один целевой элемент не взаимосвязан с по меньшей мере одним из упомянутых одного или более исходных элементов.

18. Считываемый компьютером носитель по п.17, в котором реляционная информация включает в себя набор характеристик элемента.

19. Считываемый компьютером носитель по п.17, в котором доступ к данным в упомянутом запоминающем устройстве данных выполняют в ответ на ввод пользовательских данных, представляющих выбор для просмотра одного или более данных в упомянутом запоминающем устройстве данных, которые взаимосвязаны с

упомянутым выбранным элементом.

20. Считываемый компьютером носитель по п.19, в котором упомянутая реляционная информация, соответствующая выбранному элементу, включает в себя набор характеристик элемента, связанных с выбранным элементом, и в котором упомянутые входные данные пользователя представляют выбор для просмотра одного или более элементов в запоминающем устройстве данных, которые имеют общую одну или более характеристик элементов с выбранным элементом.

21. Считываемый компьютером носитель по п.17, в котором представление упомянутого выбранного элемента и одного или более взаимосвязанных элементов включает в себя представление пользователю по меньшей мере части упомянутой реляционной информации.

22. Компьютерная система для представления пользователю взаимосвязанных элементов в запоминающем устройстве данных, содержащая:

средство для доступа к данным в упомянутом запоминающем устройстве данных, причем упомянутое запоминающее устройство данных хранит множество элементов в соответствии со схемой данных, определяющей множество элементов и взаимозависимости между элементами, и причем по меньшей мере часть упомянутого множества элементов содержит реляционную информацию, которая позволяет определять взаимозависимости между упомянутым множеством элементов;

средство для использования упомянутой реляционной информации для определения взаимозависимости между выбранным элементом и одним или более элементами, содержащими упомянутую реляционную информацию, в запоминающем устройстве данных;

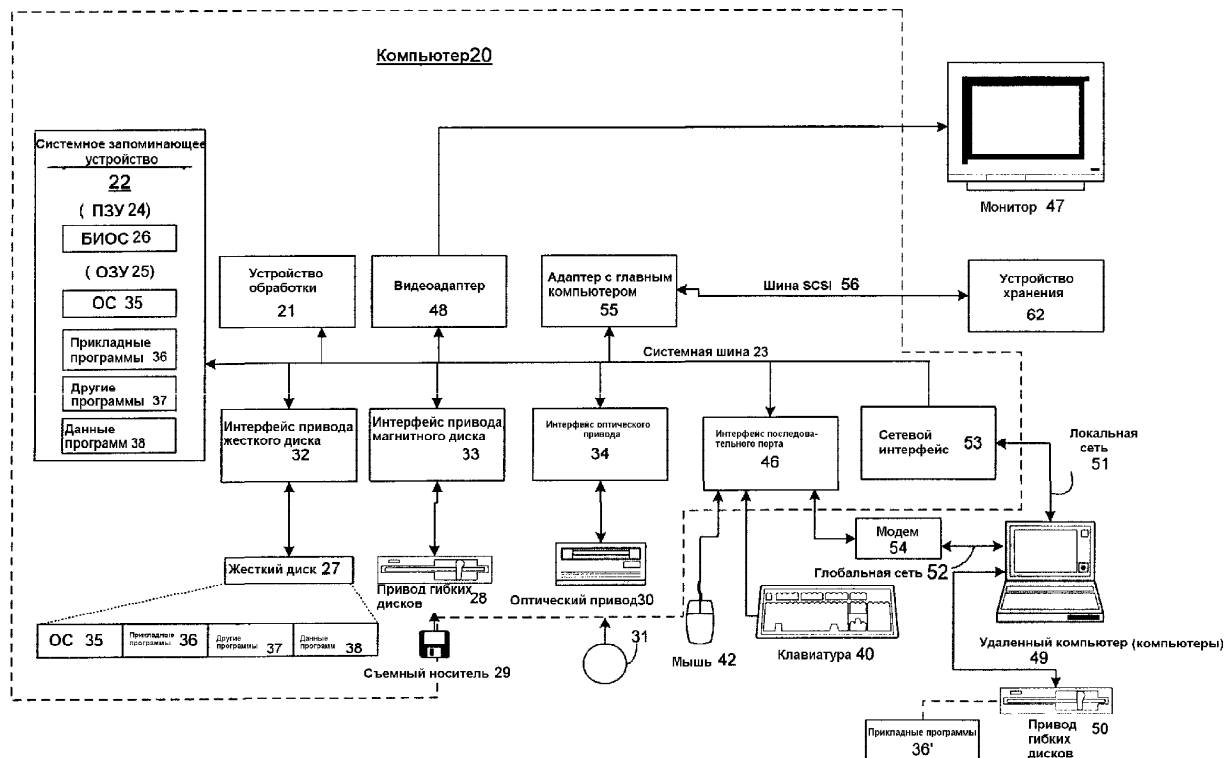
средство для отображения пользователю упомянутого выбранного элемента и одного или более взаимосвязанных элементов;

средство для приема пользовательских входных данных, вызывающих изменения в по меньшей мере одной из упомянутых взаимозависимостей, и средство для удаления по меньшей мере одного из упомянутого множества элементов из упомянутого запоминающего устройства данных в ответ на упомянутое изменение.

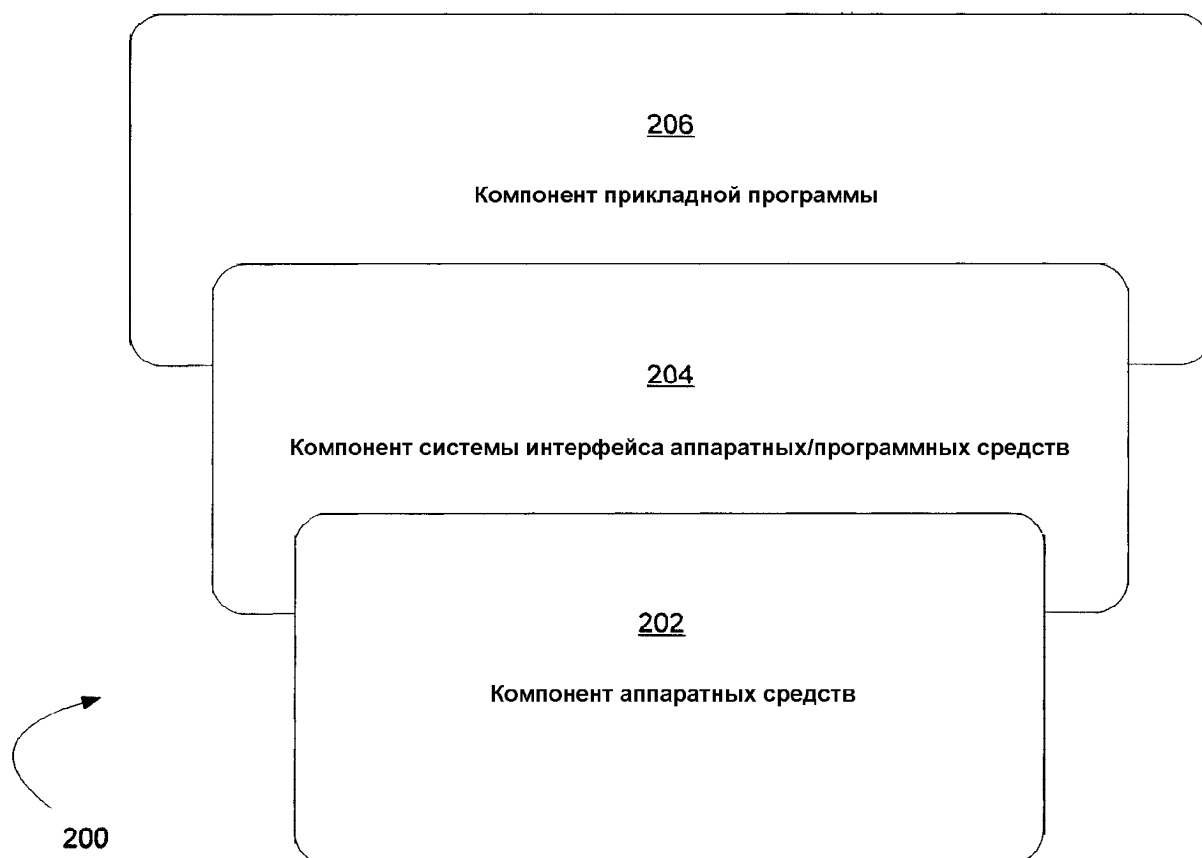
23. Компьютерная система по п.22, в которой реляционная информация, соответствующая одному или более из упомянутого множества элементов, включает в себя набор характеристик элемента.

24. Компьютерная система по п.23, в которой упомянутое средство для отображения выполнено с возможностью представления пользователю одной или более из упомянутого набора характеристик элемента.

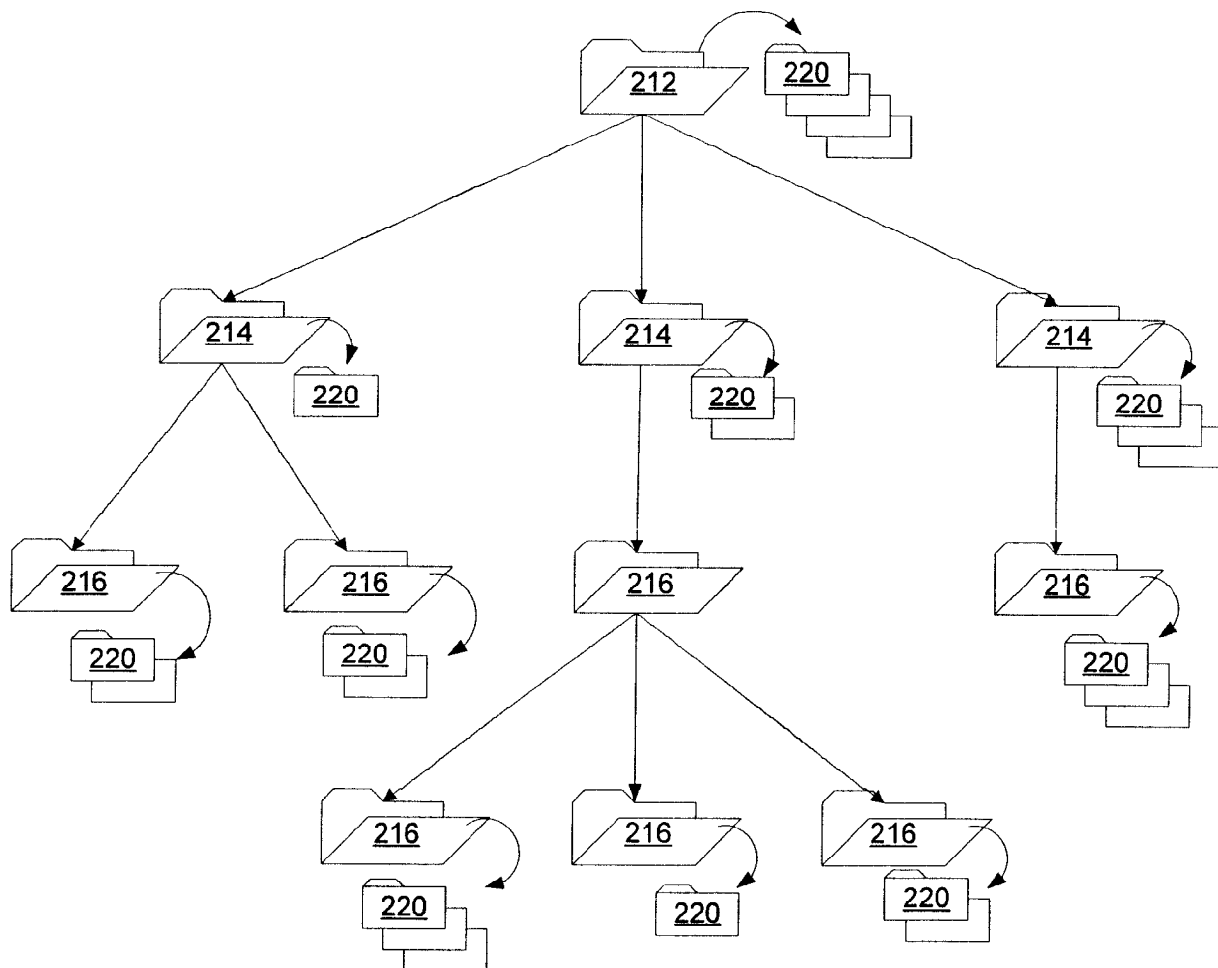
25. Компьютерная система по п.23, в которой упомянутое средство для доступа к данным в упомянутом запоминающем устройстве данных взаимодействует с упомянутым запоминающим устройством данных в ответ на пользовательский ввод данных, представляющий выбор для просмотра одного или более элементов в запоминающем устройстве данных, имеющих одну из упомянутых характеристик элемента.



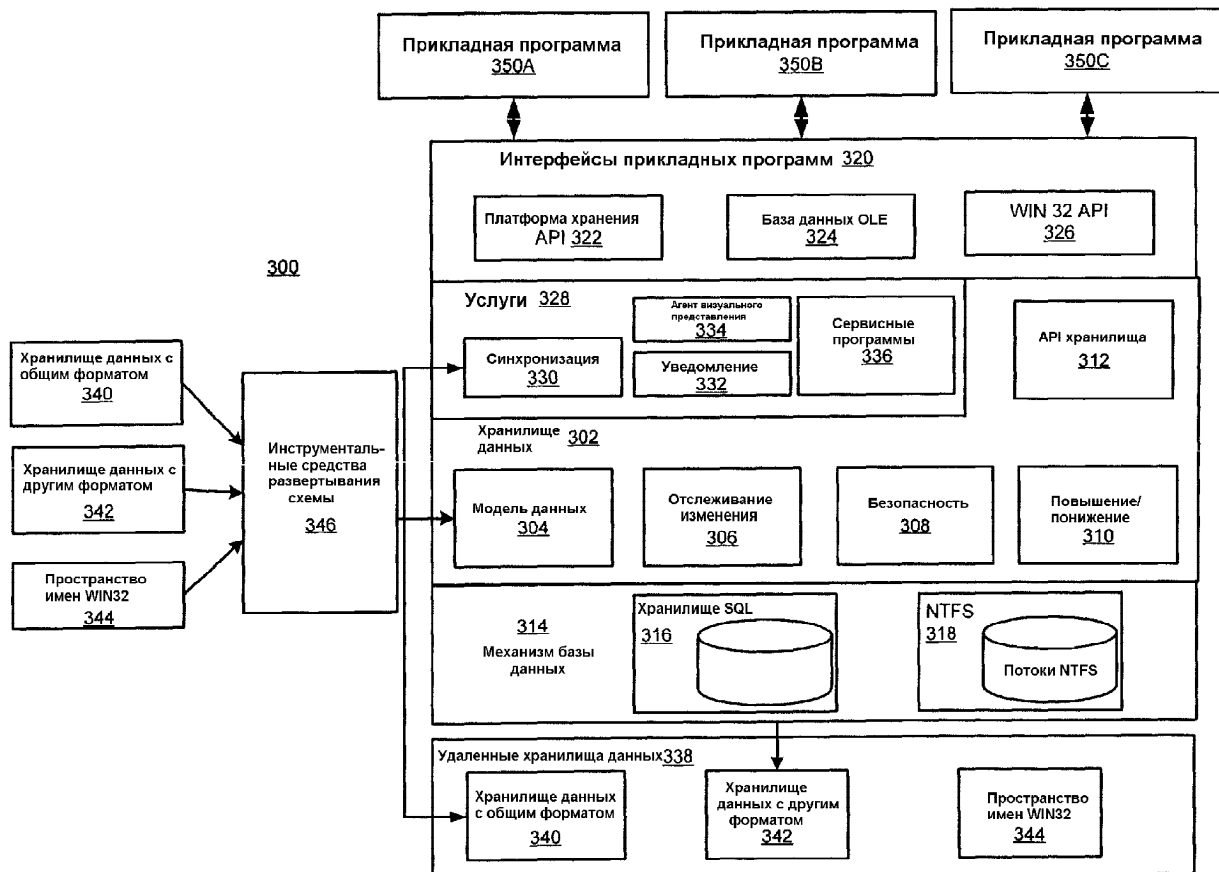
Фиг.1



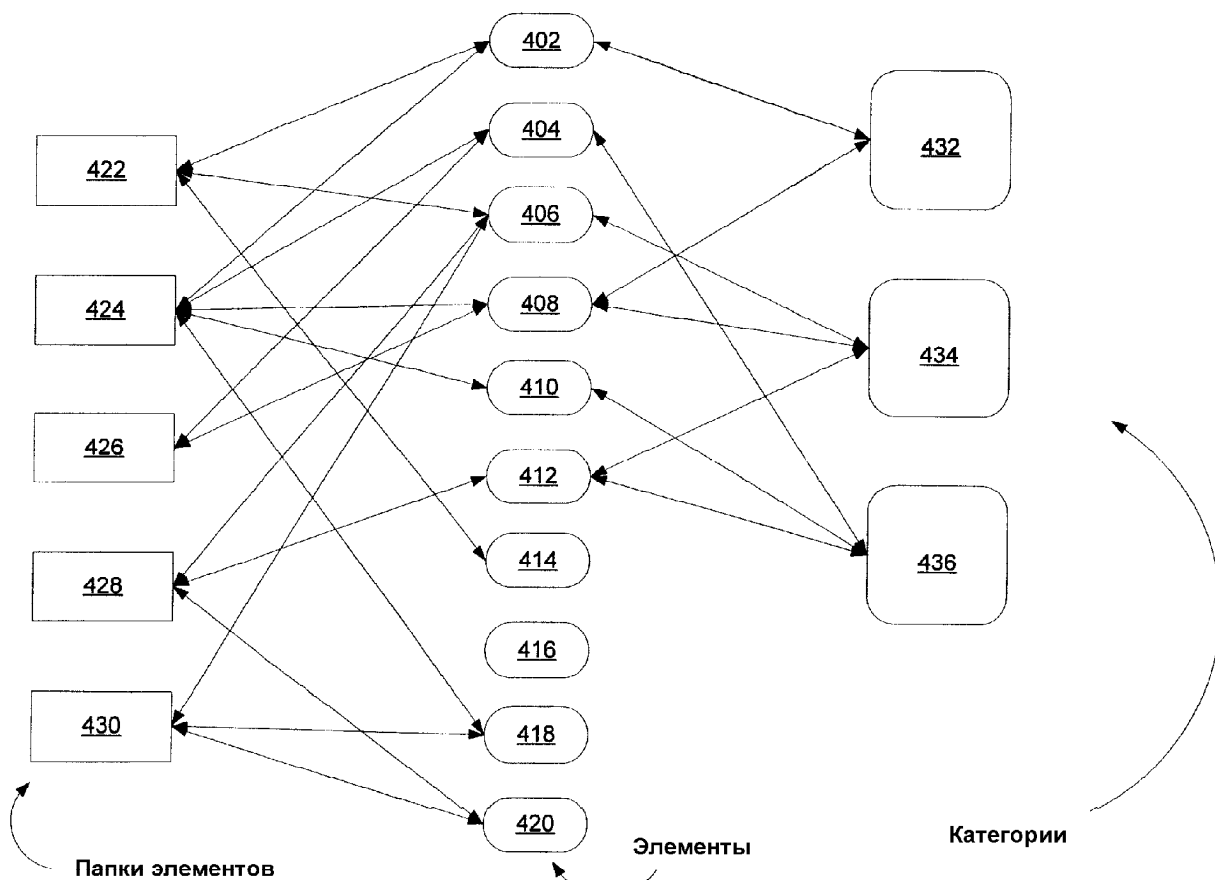
Фиг.2А



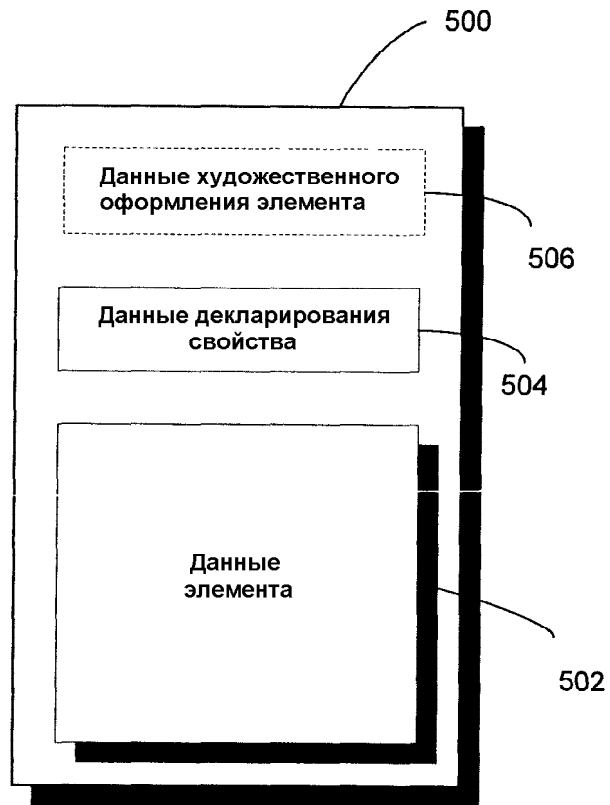
Фиг.2В



Фиг.3



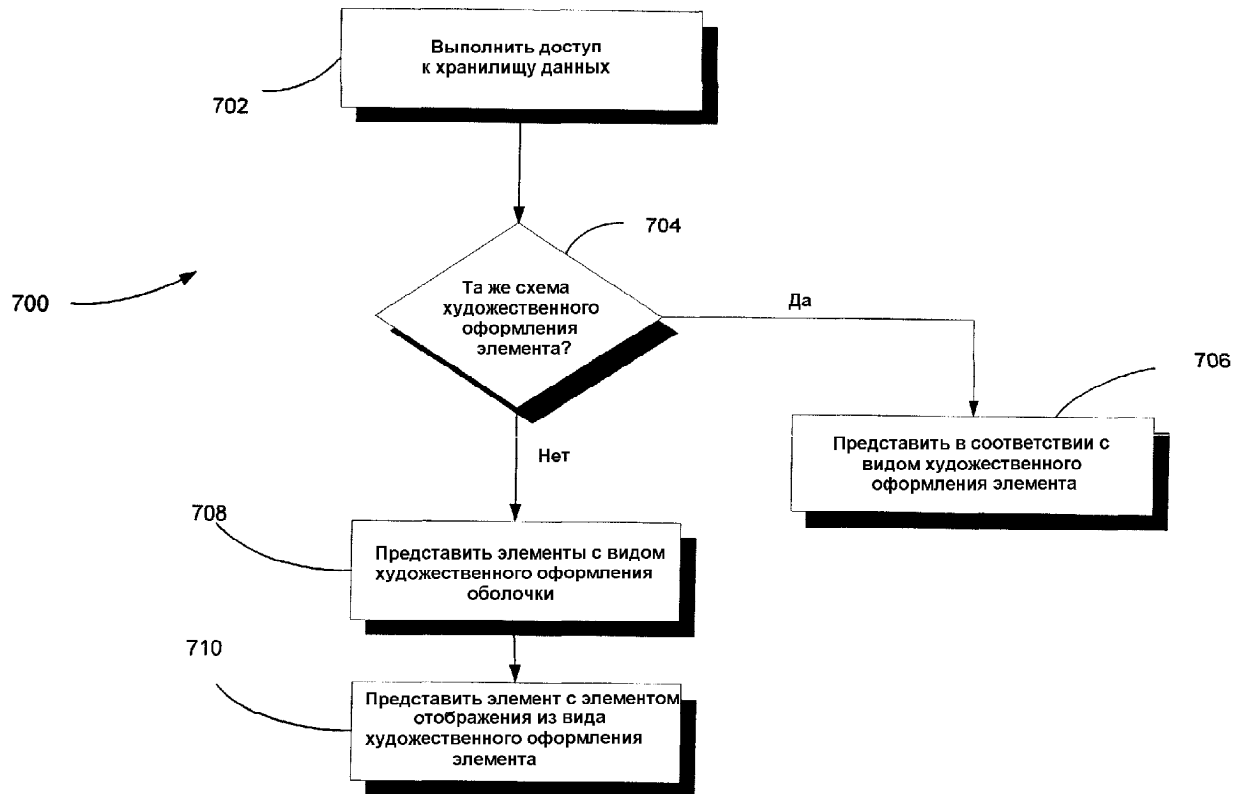
Фиг.4



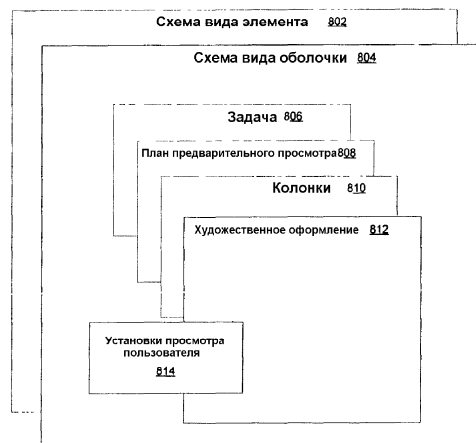
Фиг.5



Фиг.6

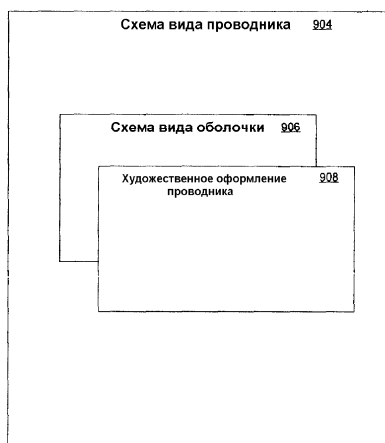


Фиг.7



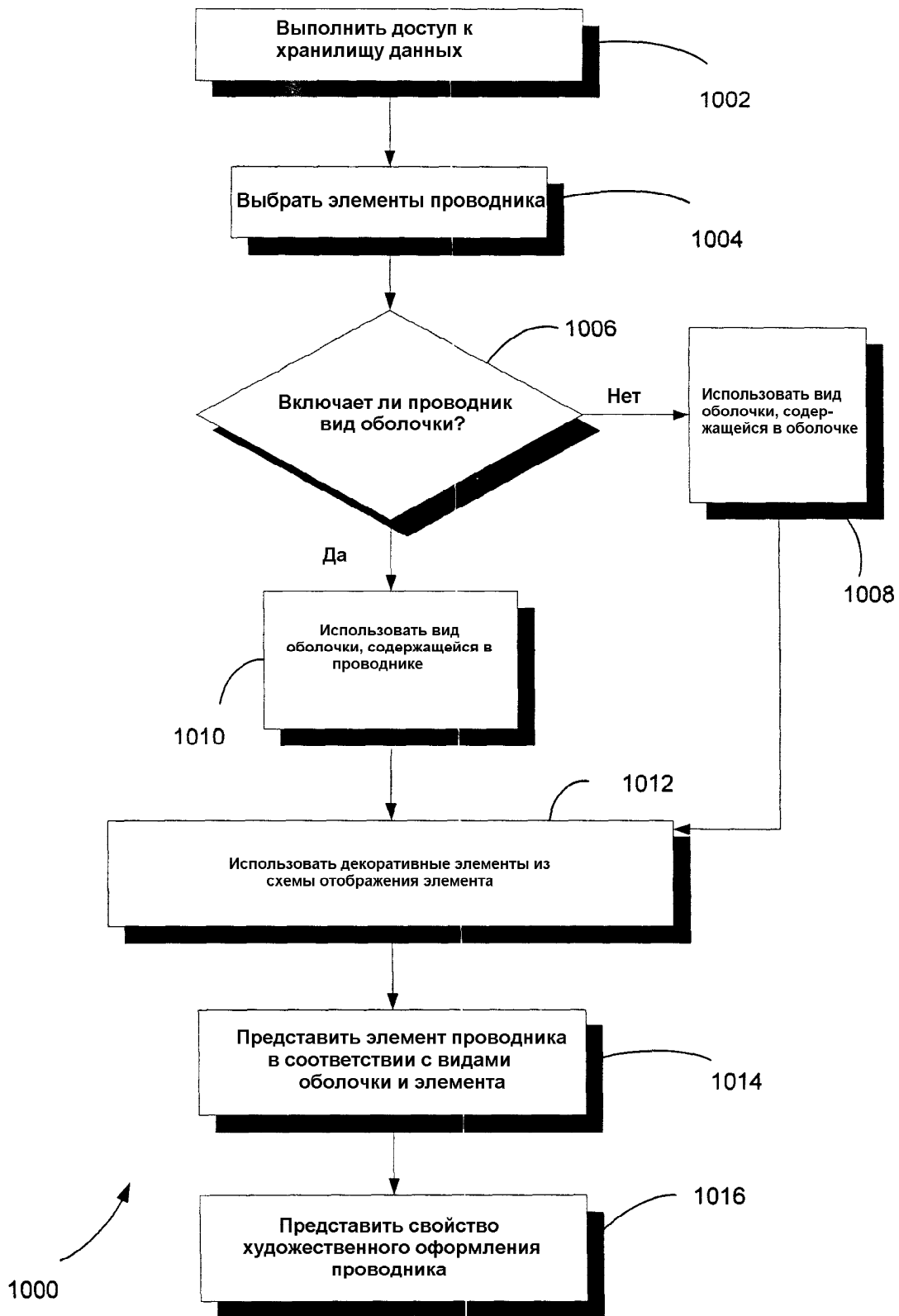
Фиг.8

Схема вида элемента 902

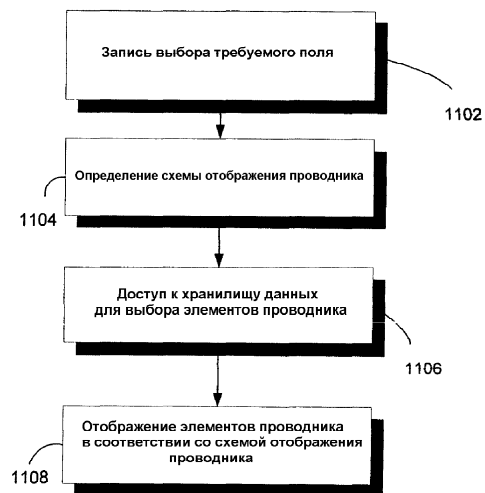


900

Фиг.9



Фиг.10



Фиг.11

1100