



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

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ(21), (22) Заявка: **2005101735/09, 25.01.2005**(24) Дата начала отсчета срока действия патента:
25.01.2005(30) Конвенционный приоритет:
26.01.2004 US 10/764,738(43) Дата публикации заявки: **10.07.2006**(45) Опубликовано: **27.09.2009** Бюл. № 27(56) Список документов, цитированных в отчете о
поиске: **US 6578032 B1, 10.06.2003. RU 2177174 C1,
12.05.2000. US 2003/0014421 A1, 16.01.2003. US
6415282 B1, 02.07.2002. US 6122405 A,
19.09.2000.**

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

(72) Автор(ы):

**ТУРСКИ Анджей (US),
ЧЕНГ Лили (US),
МАКЛОРИН Мэттью (US),
РАШИД Ричард Ф. (US)**

(73) Патентообладатель(и):

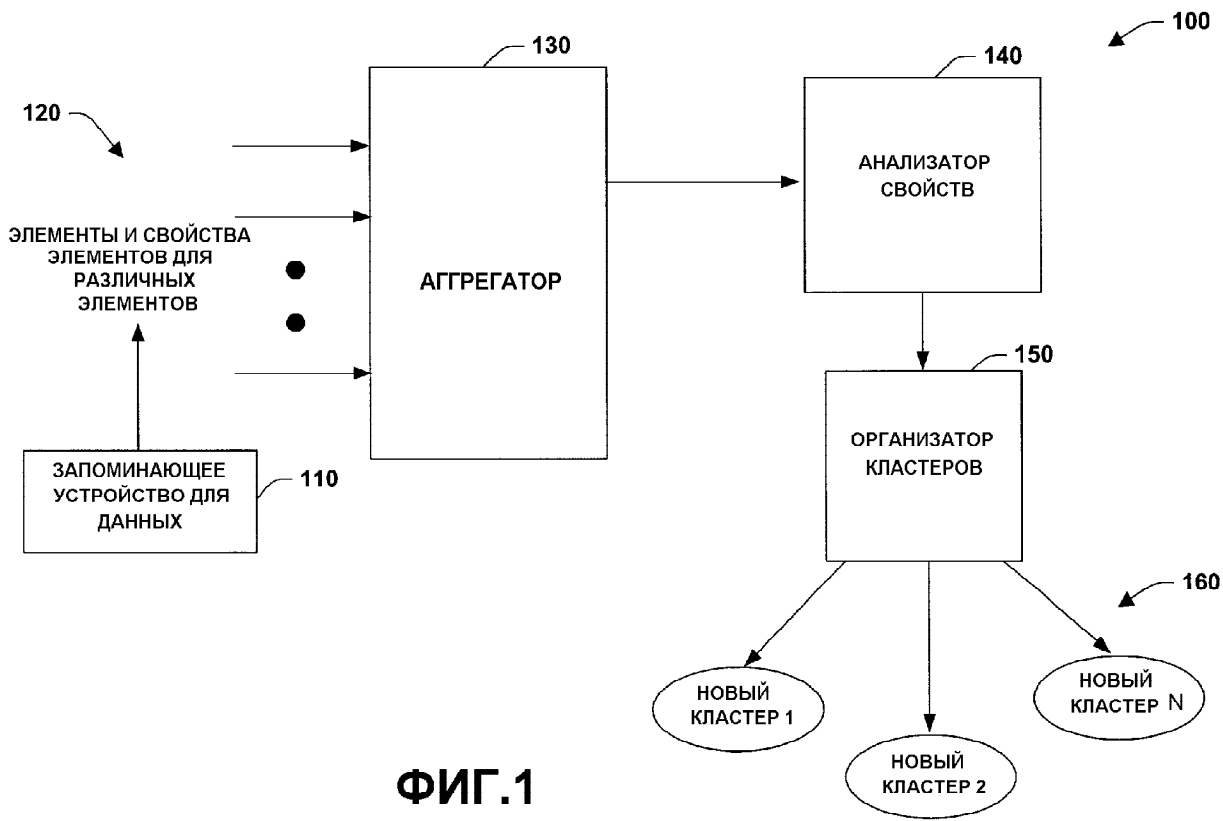
МАЙКРОСОФТ КОРПОРЕЙШН (US)

(54) АВТОМАТИЧЕСКАЯ КЛАСТЕРИЗАЦИЯ ЗАПРОСОВ

(57) Реферат:

Настоящее изобретение относится к системе и способу для автоматической кластеризации и отображения элементов данных в локальной или удаленной системе с базой данных. Изобретение позволяет упростить поиск и извлечение данных. Кластеризация может быть основана на свойствах, связанных с элементами данных, такими как тип, месторасположение, люди, дата, время, свойство, определенное пользователем, и так далее, причем исходное свойство может быть применено, чтобы образовать первый уровень кластеризации, а следующее свойство может быть определено

автоматически, чтобы образовать оптимизированную кластеризацию, из которой нужно найти и извлечь требуемую информацию. Обеспечивается компьютеризированный интерфейс для упорядочивания и извлечения данных. Интерфейс включает анализатор свойств, чтобы определять распределение элементов по меньшей мере для двух свойств кластеров, и организатор, который образует новые кластеры, частично основываясь на распределении элементов. 4 н. и 20 з.п. ф-лы, 12 ил.



ФИГ.1

RU 2368948 C2

RU 2368948 C2



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY,
PATENTS AND TRADEMARKS

(51) Int. Cl.
G06F 17/30 (2006.01)

(12) ABSTRACT OF INVENTION

(21), (22) Application: **2005101735/09, 25.01.2005**

(24) Effective date for property rights:
25.01.2005

(30) Priority:
26.01.2004 US 10/764,738

(43) Application published: **10.07.2006**

(45) Date of publication: **27.09.2009 Bull. 27**

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):

**TURSKI Andzhej (US),
ChENG Lili (US),
MAKLORIN Meht'ju (US),
RASHID Richard F. (US)**

(73) Proprietor(s):

MAJKROSOFT KORPOREJShN (US)

(54) AUTOMATIC QUERY CLUSTERING

(57) Abstract:

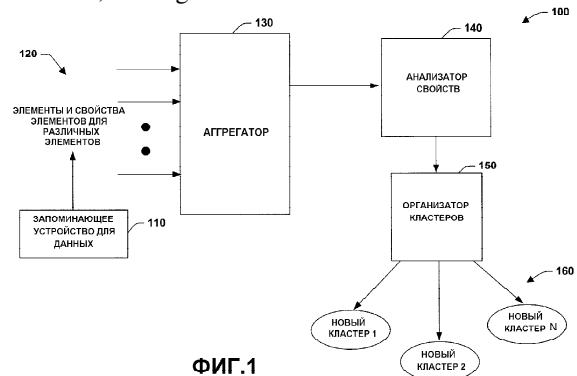
FIELD: information technology.

SUBSTANCE: present invention relates to a system and method for automatic clustering and displaying data elements in a local or remote system with a data base. Clustering can be based on properties related to data elements, such as type, location, people, date, time, user defined property etc. The initial property can be used to create the first clustering level, and the next property can be defined automatically, so as to optimise clustering, from which the required information should be searched and extracted. There is a computerised interface for sorting and extracting data. The interface can include a property analyser so as to determine distribution of elements for at least two cluster properties, and an organiser, which

creates new clusters, partially based on distribution of elements.

EFFECT: invention simplifies search and extraction of data.

24 cl, 12 dwg



ФИГ. 1

RU 2 368 948 C2

RU 2 368 948 C2

ОБЛАСТЬ ТЕХНИКИ, К КОТОРОЙ ОТНОСИТСЯ ИЗОБРЕТЕНИЕ

Настоящее изобретение относится, в общем, к вычислительным системам и, более подробно, к системе и способам, которые автоматически упорядочивают элементы информации в меньшие подмножества элементов, анализируя распределение элементов, связанное с различными кластерами свойств.

УРОВЕНЬ ТЕХНИКИ

Одним ключевым объектом операционной системы, основанной на базе данных, является способность быстро найти требуемые элементы выполнением запроса, который может включать некоторое количество свойств элементов. Следует сравнить это с предшествующими системами, которые требовали знание о месторасположении файла в иерархии папок, чтобы, например, извлечь требуемую информацию. Так как подход запроса очень эффективен, успех более новых систем зависит от способности создавать интерфейс пользователя (UI), который позволяет запросам быть простыми и наглядными для средних пользователей. Запросы баз данных (например, выраженные на языке T-SQL) в их собственной форме трудно контролировать профессиональным программистам и они обычно не подходят для конечных пользователей.

Одним подходом к проблеме запроса является проявление команд интерфейса пользователя, которые обеспечивают прямой доступ к некоторому количеству predetermined запросов. Например, может быть обеспечен predetermined запрос, чтобы найти все файлы изображений на диске (Библиотека Изображений) или всю непочитанную электронную почту. Кроме того, система может предположить группирование результатов определенным образом, например, изображения могут быть автоматически переложены в группы согласно дате их съемки. Такие растры predetermined запросов полезны для многих обычных сценариев, но они недостаточно общие, чтобы раскрыть полную силу базы данных. При использовании примера с изображениями может случиться так, что все изображения были сняты в один день (или, возможно, не были установлены часы камеры), и в этом случае группирование по дате бесполезно. Ситуация даже хуже, когда имеешь дело со свойствами третьих лиц (определенными приложением, определенными администратором или определенными пользователем). Так как эти свойства не известны создателям операционной системы, проектирование predetermined запросов для этих свойств может быть почти невозможным.

Другой подход - это обеспечить для пользователей способность запрашивать базы данных при помощи текстовых запросов, которые выглядят как естественный язык. Такие запросы могут быть достаточно общими, с точки зрения баз данных, и легкими для понимания пользователей. Тем не менее, если позволить запросы на естественном языке, которые могут принять полностью свободную форму, трудно создать программу синтаксического анализа, которая будет правильно понимать намерения пользователя в каждом случае. Если наложены некоторые грамматические ограничения, для пользователя станет труднее образовать синтаксически правильный запрос, который может иногда быть составлен порциями выражений. Так или иначе, сама идея, что текст запроса необходимо написать, не может быть привлекательной для многих пользователей. Маленькие дети, неанглоговорящие пользователи и пользователи устройств без клавиатуры (например, Планшетного Персонального Компьютера) все могут иметь проблемы с текстовым набором. Таким образом, необходим интерфейс запроса, который имеет простоту указания и выбора для нахождения и извлечения информации.

РАСКРЫТИЕ ИЗОБРЕТЕНИЯ

Нижеследующее представляет упрощенную сущность изобретения, чтобы обеспечить основное понимание некоторых объектов изобретения. Эта сущность не является подробным раскрытием изобретения. Она не предназначена, чтобы 5
устанавливать ключевые/критические элементы изобретения или очерчивать объем изобретения. Ее единственная цель - представить некоторые понятия изобретения в упрощенной форме как вступление к более подробному описанию, которое представлено позже.

Настоящее изобретение относится к автоматическому поиску и отображению 10
требуемой информации в подмножестве легкоуправляемых кластеров информации. В файловой системе интерфейса пользователя передвижение большого набора элементов, такого как при отображении элементов в виде списков, становится проблематичным при попытке найти и извлечь требуемую информацию из таких 15
списков. Настоящее изобретение обеспечивает улучшенный указательный интерфейс, который облегчает передвижение большого набора элементов, классифицируемых по связанным с ними свойствам элементов. Элементы, объединенные в кластеры по этим свойствам, могут быть представлены в виде папок (или другим типом отображения), посредством чего может быть выполнена автоматическая кластеризация по 20
различным или следующему свойству, чтобы разделить или упорядочить результаты запроса в легкоуправляемые подмножества кластеров. Эти подмножества могут затем быть выбраны, чтобы извлечь требуемую информацию или выполнить другие процедуры кластеризации (например, вложенную кластеризацию). Лучшее свойство для образования кластера может быть определено, анализируя распределение 25
элементов в различных кластерах свойств.

Один объект настоящего изобретения обеспечивает автоматический выбор свойства кластеризации. Чтобы определить такие свойства, ставится следующая задача: заданы 30
исходный набор элементов и набор свойств элементов, которые могут быть использованы для группирования, какое свойство, связанное с набором элементов, предлагает лучшие результаты кластеризации? Посредством лучших результатов кластеризации настоящее изобретение пытается обеспечить единообразное группирование результатов в небольшое количество кластеров. Таким образом, 35
случаи, когда существует всего несколько кластеров с большим количеством элементов или большое количество кластеров с всего несколькими элементами в каждом кластере, обычно не желательны, чтобы эффективно находить и извлекать требуемую информацию.

Вышеуказанная задача может быть решена, задавая оценку кластеризации для 40
каждого свойства элемента и выбирая свойство с наивысшей оценкой. Оценка кластеризации может быть вычислена перемножением вместе числа элементов в каждом кластере. Для N элементов функция для вычисления оценки кластеризации как произведения размеров кластера имеет свой максимум, когда элементы разбиты на \sqrt{N} 45
кластеров, соответствующих кластеров, имеющих \sqrt{N} элементов. Для других распределений оценка используется, чтобы измерить и сравнить, насколько далеко распределение от идеального распределения. Пример альтернативной функции оценки может быть основан на биномиальном распределении, например. Для таких типов 50
распределений значение оценки имеет статистическую интерпретацию, которая обеспечивает некоторое количество способов, которыми N -суммарных элементов могут быть разбиты на кластеры заданного размера. Кластеризация, которая имеет наибольшее значение для пользователя, - это та, которая уменьшает наибольшее количество альтернативных распределений. Чтобы сравнить различные свойства,

которые могут быть использованы для последующей кластеризации, оценки кластеризации могут быть вычислены для всех свойств, причем такие вычисления могут быть легко выполнены одним проходом по списку элементов.

5 Для достижения вышеупомянутого и связанных с ним целей, определенные иллюстративные объекты изобретения описаны здесь в связи с последующим описанием и приложенными чертежами. Эти объекты служат признаком различных способов, которыми изобретение может быть осуществлено на практике, все из которых замышлены, чтобы охватываться настоящим изобретением. Другие
10 предпочтения и новые признаки изобретения могут проявиться в последующем подробном описании изобретения, когда оно рассматривается в объединении с чертежами.

КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ

15 Фиг.1 - это схематическая блок-схема системы кластеризации в соответствии с объектом настоящего изобретения.

Фиг.2 - это блок-схема, иллюстрирующая процесс автоматической кластеризации запросов в соответствии с объектом настоящего изобретения.

20 Фиг.3-10 иллюстрируют пример интерфейсов пользователя для автоматической кластеризации запросов в соответствии с объектом настоящего изобретения.

Фиг.11 - схематическая блок-схема, иллюстрирующая подходящую операционную среду в соответствии с объектом настоящего изобретения.

25 Фиг.12 - это схематическая блок-схема примерной вычислительной среды, с которой настоящее изобретение может взаимодействовать.

ОСУЩЕСТВЛЕНИЕ ИЗОБРЕТЕНИЯ

Настоящее изобретение относится к системе и методологии для автоматической кластеризации и отображения элементов данных в локальной или удаленной системе с базой данных. Такая кластеризация может быть основана на свойствах, связанных с
30 элементами данных, такими как тип, месторасположение, люди, дата, время, свойства, определенные пользователем, и так далее, причем исходное свойство может быть определено автоматически, чтобы образовать оптимизированную кластеризацию, из которой нужно найти и извлечь требуемую информацию. В одном объекте обеспечивается компьютеризированный интерфейс для упорядочивания и извлечения
35 данных. Интерфейс включает анализатор свойств, чтобы определять распределение элементов по меньшей мере для двух свойств кластера, и организатор, который образует новые кластеры, частично основываясь на распределении элементов.

Использующиеся в этом приложении термины «компонента», «анализатор»,
40 «кластер», «система» и тому подобное предназначены, чтобы ссылаться на объект, связанный с применением вычислительной машины: или аппаратные средства, комбинацию аппаратных и программных средств, программные средства или выполняющиеся программные средства. Например, компонента может быть, но не ограничивается, выполняющимся процессором процессом, процессором, объектом,
45 исполняемым файлом, потоком управления, программой и/или компьютером. В качестве иллюстрации компонентами могут быть как приложение, выполняющееся на сервере, так и сервер. Одна или более компонент могут постоянно храниться в процессе и/или потоке управления, и компонента может быть расположена на одном
50 компьютере и/или быть распределенной между двумя или более компьютерами. Также эти компоненты могут выполняться на различных считываемых компьютером носителях, имеющих различные, хранящиеся на них структуры данных. Компоненты могут взаимодействовать via локальные и/или удаленные процессы, например, в

соответствии с сигналом, имеющим один или более пакетов данных (например, данные с одной компоненты, взаимодействующей с другой компонентой в локальной системе, распределенной системе и/или через сеть, такую как Интернет, с другими системами via сигнал).

5 Первоначально ссылаясь на Фиг.1, система 100 кластеризации запросов иллюстрируется в соответствии с объектом настоящего изобретения. Система 100
10 включает запоминающее устройство 110 для данных, которое хранит множество элементов 120 данных, которые должны быть отображены на интерфейсе пользователя (не показано). Такие элементы 120 могут включать документы, файлы, папки, изображения, аудиофайлы, исходный код и так далее, которые могут
15 появляться в различных видимых состояниях на интерфейсе пользователя, который ниже описан более детально. Элементы 120 также связаны с различными свойствами (например, метаданными), описывающими такие объекты, как тип элемента (например, изображение, документ, крупноформатная таблица, двоичный код и так
20 далее), дату создания, связанных с элементом людей, месторасположение, категорию, свойство, определенное пользователем, и так далее. Агрегатор 130 собирает элементы 120 и связанные с ними свойства и представляет элементы на анализатор
25 140, который выполняет анализ соответствующих элементов и свойств. Например, такой анализ может включать автоматическое определение оценки для различных возможных сценариев кластеризации или потенциального группирования элементов.

Основываясь на анализе анализатора 140, организатор 150 кластеров представляет
30 оптимизированное группирование новых кластеров 160 для пользователя. Оптимизированное группирование кластеров 160 облегчает поиск и извлечение требуемой информации из запоминающего устройства 110 для данных, которое может
35 включать локальные запоминающие среды, удаленные запоминающие среды или комбинацию локального и удаленного запоминающих устройств.

В одном примере автоматической кластеризации активное окно кластеризации по
40 умолчанию может группировать элементы по типу элемента. При изучении пользователей было обнаружено, что группирование первого уровня по типу элемента полезно и хорошо понимается пользователями. Тем не менее, было также
45 обнаружено, что второй уровень кластеризации по другому свойству не очевиден и труден для обнаружения. Таким образом, одним объектом настоящего изобретения является автоматический выбор свойства кластеризации. Задача ставится следующим
50 образом: даны исходный набор элементов и набор свойств элементов, которые могут быть использованы для группирования, какое свойство предлагает лучшие результаты кластеризации? Посредством лучших или оптимизированных результатов кластеризации целью является обеспечить единое группирование элементов в
55 небольшое количество кластеров.

Вышеуказанная цель может быть достигнута задаванием оценки кластеризации для
каждого свойства элемента и выбором свойства с наивысшей оценкой. Оценка
45 кластеризации может быть вычислена перемножением вместе числа элементов в каждом кластере так, как в следующем уравнении:

$$\text{оценка} = n_{\text{пунктов в кластере 1}} * n_{\text{пунктов в кластере 2}} * \dots$$

Для N элементов функция вычисления оценки кластеризации как произведения
50 размеров кластера имеет свой максимум, когда элементы разбиты на \sqrt{N} кластеров, каждый кластер имеет \sqrt{N} элементов. Для других распределений оценка используется, чтобы померить и сравнить, насколько далеко она от идеального или оптимизированного распределения. Было обнаружено, что вышеуказанная функция

оценки дает разумные результаты в контрольных примерах. Тем не менее, отмечено, что функция оценки используется как пример. Например, могут быть применены другие функции, которые обеспечивают различный статистический вес распределений по отношению к идеальному распределению.

5 Пример альтернативной функции оценки основывается на биномиальном распределении следующим образом:

оценка = $(N_{\text{суммарное}})! / ((n_{\text{элементов в кластере 1}})! * (n_{\text{элементов в кластере 2}})! * \dots)$

10 В этом примере значение оценки имеет статистическую интерпретацию, которая обеспечивает некоторое количество способов, которыми $N_{\text{суммарных}}$ элементов могут быть разбиты на кластеры заданного размера. Кластеризация, которая имеет наибольшее значение для пользователя, - это та, которая уменьшает наибольшее число альтернативных распределений. Чтобы сравнить различные свойства, которые
15 могут быть использованы для последующей кластеризации, вычисляются оценки кластеризации для всех свойств. Это может быть легко достигнуто одним проходом по списку всех элементов, как описано более подробно в процессе, начерченном на Фиг.2.

Фиг.2 - это блок-схема, иллюстрирующая процесс 200 автоматической
20 кластеризации в соответствии с объектом настоящего изобретения. Несмотря на то, что для упрощения объяснения методология показывается и описывается как серия действий, необходимо понимать и оценить, что настоящее изобретение не ограничивается последовательностью действий, так как некоторые действия могут в соответствии с настоящим изобретением происходить в различной
25 последовательности и/или одновременно с другими действиями в отличие от того, как показано и описано здесь. Например, специалисты в данной области техники понимают и ценят, что методология может альтернативно быть представлена как серия взаимосвязанных положений или событий, таких как положения на диаграмме. Кроме того, не все проиллюстрированные действия могут потребоваться, чтобы
30 реализовать методологию в соответствии с настоящим изобретением.

Допустим, существует N элементов и M свойств, которые нужно сравнить, процесс 200 может быть применен следующим образом: на этапе 210 инициализируют M хэш-таблиц. На 220 выполняют итерацию по N элементам. На
35 этапе 230 для каждого элемента проводят итерацию по M свойствам. На этапе 240 для каждого свойства элемента вычисляют значение хэш-функции. Хэш-функция выбирается таким образом, чтобы два значения свойства, проходящих через один и тот же кластер, возвращались с одинаковым значением хэш-функции. Например, при
40 кластеризации по дате/времени хэш-функция может быть основана только на части, связанной с датой, игнорируя временную часть. На этапе 250 применяют хэш-таблицы, чтобы отследить число кластеров и число элементов в каждом кластере. На 260 вычисляется оценка кластеризации для каждого свойства, используя данные из связанной с ними хэш-таблицы.

45 На этапе 270 свойства в списке упорядочиваются по качеству кластеров, которое они могут произвести. Если число элементов превысит некоторую пороговую величину (например, более 10 элементов), результаты могут быть автоматически объединены в кластеры, используя верхнее свойство списка на этапе 280. Также могут
50 быть предложены другие кластеры, которые являются следующими по порядку альтернативами. Например, при выборе всех элементов типа email message вышеуказанный процесс автоматически объединит в кластеры результаты по message sender в контрольных примерах для сообщений электронной почты. Тем не менее,

выбирая, например, элементы типа Word Document, были созданы кластеры, основанные на last modification data, тогда как элементы типа C# source files были сгруппированы по содержанию их папок (что соответствует группированию по проекту программирования). Общая природа вышеуказанных подходов позволяет
5 определить алгоритм группирования, который наиболее подходит для данного набора элементов, включенных в вычисление обычных свойств, а также свойств третьих лиц.

Фиг.3-10 иллюстрируют различные примеры интерфейсов пользователей, которые
10 иллюстрируют одну или более систем автоматической кластеризации и описанные ранее процессы. Отмечено, что эти интерфейсы могут включать дисплей, имеющий один или более объектов отображения, причем объекты отображения включают такие объекты, как реконфигурируемые пиктограммы, кнопки, ползунки, окна для ввода, варианты выбора, меню, ярлыки и так далее, имеющие многочисленные
15 реконфигурируемые размеры, формы, цвета, текст, данные и звуки, чтобы облегчить действия системы 100. Дополнительно, интерфейс может также включать множество других входных данных и управляющих элементов для регулировки и конфигурирования одного или более объектов настоящего изобретения, как будет
20 описано более подробно ниже. Это может включать получение команд пользователя с мыши, клавиатуры, речевого ввода, узла всемирной паутины, удаленной службы всемирной паутины и/или другого устройства, такого как камера или входной видеосигнал, чтобы воздействовать или видоизменять действия интерфейса или других объектов системы 100.

Следующее рассмотрение описывает различные объекты настоящего изобретения и
25 относится к примерным интерфейсам, изображенным на Фиг.3-10. При проектировании папки или структуры другого типа проектировщики (как программист приложения, так и конечный пользователь) имеют высокую степень
30 свободы, которая позволяет скрыть неважные или редко используемые элементы из вида активного окна перемещением элементов в скрытый каталог. Подобным образом при создании программы просмотра, основанной на свойствах, могут быть обеспечены различные механизмы, чтобы скрывать свойства, которые бессмысленны или же не очень полезны, даже если алгоритм кластеризации определяет их высокую
35 оценку.

Свойство повышения/понижения качества может быть рассмотрено на различных
уровнях. На уровне приложения проектировщик приложения может указать, какие
40 первичные свойства должны быть проявлены на интерфейсе пользователя, а какие являются вторичными или вспомогательными. Обычно это должно быть определено отдельно для каждого типа элемента. Автоматическая кластеризация запроса, описанная в предыдущей части, обычно рассматривает первичные свойства. Кроме того, каждый тип элемента должен определить свойство, отображающее все свойства, которые обычны для элементов. Например, обычное свойство Date может быть
45 отображено как свойство Date Taken для изображений, но как Last Modification Date для документов. Подобным образом поле People может быть Author для документов, но Sender для электронной почты и так далее.

Обычно существует пользователь, который должен уметь решать, какие свойства
50 являются лучшими, чтобы показывать соответствующие им данные. Может существовать явный UI, чтобы выдвинуть или удалить частное свойство, но настоящее изобретение также может учиться косвенным образом на действиях пользователя (например, via обучающие алгоритмы). Каждое свойство может иметь свой вес,

который возрастает, когда пользователь переключается с кластеризации по одному свойству на кластеризацию по другому, и уменьшается при обратном переключении. Конечный рейтинг каждого свойства (используется, чтобы решить, по какому свойству образовывать кластеры) является произведением веса свойства и оценки кластеризации (вычисленной согласно формулам, описанным выше).

Как было рассмотрено выше, пользователи обычно предпочитают иерархическую организацию item type кластеров в однородном списке. Иерархия представляет несколько типов упорядочивания и делает простым поиск требуемого item type значения. Это должно быть верным для любого свойства, которое имеет более чем несколько различных значений свойств. Нижеследующее описывает конкретный пример технологии для упорядочивания значений свойств в иерархическом виде. В случае стандартных файлов тип элемента определяется по расширению файла. Дружественные к пользователю имена для типов файлов могут быть использованы как определяемые текущей программой просмотра. Различные расширения файлов, которые имеют результатом одинаковые дружественные имена, обычно уже сгруппированы вместе (например, и.h и.hxx называются C/C++ Header Files). Дополнительно, может быть представлен еще один уровень иерархии группированием всех файлов похожего типа. В прототипе были рассмотрены и обработаны метагруппы Document Files, Picture Files, Music and Video Files, Programming Files и Other Files. Также метагруппы людей могут быть обработаны как объекты класса.

Например, список item type=people может быть разбит на меньшие части по типу канала связи, который может быть использован, чтобы связаться с данным человеком. Эти части включают группы людей, с которыми можно связаться через почту, по телефону, мгновенной передачей сообщений или через электронную почту, например. Каждую из этих групп, если требуется, можно поделить далее. Например, в корпоративной среде адреса электронной почты могут быть поделены на внутренние (извлеченные из корпоративной адресной книги) и внешние (обычно из персонального контактного списка пользователя). Некоторые люди могут иметь многочисленные способы связи, в каждом случае они могут оканчиваться в многочисленных кластерах. Кластеры свойств в отличие от традиционных каталогов не имеют ограничений на то, что элемент может находиться только в одном месте.

Каталоги представляют группирование элементов, созданное пользователем. Несмотря на то, что ожидается, что со временем эта кластеризация элементов, основанная на свойствах, уменьшит необходимость и важность каталогов, каталоги все еще поддерживаются. Каталоги обычно иерархически упорядочиваются, и кластеры каталогов должны иметь сходство с этой иерархией. Одним неудобством иерархии каталогов является то, что она включает некоторое количество директорий, представляющих небольшой интерес для пользователя, такие как Program Files или директория Windows. При использовании существующих каталогов, чтобы упорядочить элементы в кластеры, очевидным усовершенствованием является отображать только часть иерархии каталогов, которая содержит некоторые из элементов на виду.

Фиг.3 - это примерный интерфейс 300, который содержит программные файлы (на Томе С:). В Windows Explorer, например, вид включает полную структуру папок. В прототипе кластеризация файлов по «категории» включает только папки, относящиеся к фактически выбранному набору элементов (подмножеству полного дерева папок).

Фиг.4 - это интерфейс 400, который демонстрирует кластеризацию по папкам.

Другим объектом иерархии папок является то, что она объединяет понятие физического расположения (этот или тот диск, или внешний совместно используемый ресурс) с понятием логического расположения (размещения в иерархии папок). Так как могут быть созданы логические группы, которые могут охватывать несколько физических расположений, физическое расположение может быть отделено от свойства папки и таким образом может представлять папки, имеющие одинаковое имя, вместе, независимо от их физического расположения. Может быть оценено, что также обеспечивается группирование по месторасположению.

Фиг.5 является примерным интерфейсом 500 папки (VSS), которая существует на двух накопителях (Том С: и Том D:). При просмотре «категории» VSS интерфейс 500 объединяет содержание папки с ее физических расположений на позицию 510. Эта функциональная возможность основывается на предположении, что если две или более папок имеют одинаковое имя, это случается нарочно. Если не так, то файлы могут быть легко отделены по свойству 610 расположения в интерфейсе 600 пользователя, изображенном на Фиг.6.

Фиг.7 является интерфейсом 700, иллюстрирующим кластеризацию по свойствам даты. Кластеризация по дате и времени имеет естественную иерархию год/месяц/день/час/минута. Тем не менее, существует также понятие относительного времени - относительно now. Вероятно, что оба понятия важны. Кластеры дат включают некоторое количество предопределенных запросов (динамических групп), которые включают элементы из today, yesterday и так далее.

Одной интересной классификацией элементов является классификация по связанным с элементами людям. Существует много свойств элементов, которые могут быть использованы, чтобы создать такие ассоциации, например sender или recipient для сообщений электронной почты или прикреплений, author для документов, person pictured для фотографий и так далее. Кластеризация элементов по людям может ставить конкретную задачу из-за социального подтекста, переносимого представлением иерархии людей. Например, люди могут быть сгруппированы по некоторым формальным атрибутам, таким как Internal или External Contacts, но некоторые из этих групп могут все еще быть слишком большими, чтобы управлять ими эффективно. Например, список внутренних контактов, на который ссылается эталонное сообщение электронной почты, имеет около 5000 имен.

Список может быть упорядочен по алфавиту или сгруппирован по первым буквам (как в словаре), но любой список такой длины обычно трудно охватить. Одной проблемой является то, что имена людей, важных для пользователя, затмеваются именами малоизвестных людей, которые в списке случайны. Можно допустить, что большинство важных контактов - это те, которым пользователь пишет наиболее часто и последним по времени, или те, кто являются авторами или соавторами документов на диске пользователя и так далее. Используя некоторый взвешенный анализ, список людей упорядочивается по их относительной важности, которая может быть составлена для пользователя.

Тем не менее, представление списка имен людей, упорядоченного по их вычисленной важности, может не быть допустимым решением. Вычисленный порядок может случайно и неправильно отражать чье-либо чувство важности, тогда как поиск имен около середины списка или в вершине списка может все еще быть очень трудным. Должна быть использована важная информация, чтобы выбирать, чьи имена показывать первыми или в активном окне, но упорядочивать имена по алфавиту, чтобы сделать поиск конкретного имени более легким и уменьшить

возможные предположения об относительной важности людей.

Фиг.8 - это примерный интерфейс 800, который иллюстрирует полусвернутые списки для просмотра соответствующих людей. Эти списки могут включать иерархическое расширение списка людей, который, тем не менее, представлен для пользователя как отдельный однородный список, упорядоченный по алфавиту. Когда список показывается в первый раз, он содержит только несколько верхних (10-20) наиболее важных имен в алфавитном порядке. Это позволяет простым одним нажатием достичь информации о наиболее значимых людях. В то же время верхние имена действуют как закладки словаря - каждое может быть развернуто, чтобы показать имена второго уровня или других третичных уровней.

Это похоже на иерархическое расширение за исключением того, что все развернутые имена показаны в активном окне как принадлежащие первому уровню имен. Последнее обеспечивается, чтобы уменьшить скрытый смысл, что одна персона выше, чем другая, что может быть негативно воспринято, если это не следует иерархии организации, например. Расширение списка может быть продолжено, пока нижние имена в списке важности не будут видны. Тем не менее, так как расширение может быть выполнено на выбранных областях списка, суммарное число видимых имен может быть ограничено, обычно всего десятью именами. В любое время видимые имена сортируются по алфавиту и представляются как отдельный список. Это делает простым поиск требуемого имени. Отмечено, что полусвернутый список может быть применен ко многим различным классификациям, не только людям. Несколькими очевидными классификациями являются список ключевых слов (категорий) и список словарных (энциклопедических) статей.

Идея использовать существующие статьи как индексы каталога обычна. На самом деле - это стандартный способ упорядочивания печатных словарей. Тем не менее, в стандартном словарном подходе индексы вставляются в начале и в конце каждой страницы, чтобы показать содержание этой страницы. Можно описать это как «постоянный промежуток» между последовательными индексами. Слова, выбранные для индексов, в любом случае не являются особенными, они просто случайно оказываются в начале или конце станицы.

В настоящем изобретении имена, выбранные для индексов, являются верхними в списке «важности». Используя аналогию со словарем, это должны быть слова, которые наиболее часто просматриваются. Кроме того, эти имена сами являются статьями - имя выбирается нажатием на него. Это обеспечивает доступ одним нажатием к большинству обычных статей быстрее, чем прокруткой к странице, которая содержит статью. С другой стороны, между индексами может быть различное количество статей второго порядка. Когда количество статей второго порядка достаточно велико, может быть создан индекс третьего порядка и так далее.

Фиг.9 иллюстрирует полусвернутые группы 900, тогда как группа 1000 показана в развернутом состоянии на Фиг.10 при ее выборе из групп 900. Фиг.10 также изображает группу 1000 в полусвернутом состоянии 1010. При представлении кластеров (или других способов группирования элементов вместе) другой вопрос - как визуализировать кластеры на экране. Обычными способами визуализации групп является показать некоторое представление групп как целого (свернутый вид) или как совокупность всех элементов в группе (развернутый вид). В стандартном представлении Windows со списком папок слева и списком элементов справа может иметься в виду расширенный вид для видимых в настоящее время папок и свернутый вид для всех остальных папок. Подпапки текущей папки обычно показываются в

свернутом виде, даже если свернутое изображение подпапки может содержать внутри группу из нескольких элементов. Иногда более чем одна расширенная группа может быть визуализирована одновременно или когда показаны элементы, сгруппированные в пакеты.

5 В программе просмотра файлов, которая позволяет группирование и может отображать многочисленные группы одновременно, для групп обычно быть «сворачиваемыми» - содержание групп может быть индивидуально показано или скрыто. Тем не менее, группа все еще существует в двух состояниях, и развернутое
10 состояние позволяет взаимодействовать с индивидуальными элементами в группе. В случае больших групп расширение одной группы затмевает видимость всех остальных, что делает вид нескольких групп не слишком полезным.

В настоящем изобретении представлено третье состояние, которое показывает несколько первых элементов группы, так называемое «сжатое» или «полусвернутое»
15 состояние 900 группы. Чтобы осуществлять цикл между развернутым состоянием 1000, сжатым 1010 и свернутыми состояниями 900, повторно нажимается отдельная кнопка. Интерфейс 900 является Программой Просмотра Файлов, показывающей две полусвернутые группы и третью, достаточно маленькую, чтобы быть полностью
20 показанной позицией 910.

Предпочтение сжатого состояния в том, что группа занимает меньше места на экране, чем в открытом состоянии, но дает пользователю больше информации о группе, чем закрытое состояние. Это позволяет визуализировать большее число групп, все еще обеспечивая подробную информацию о содержании групп. Пользователь
25 может быстрее определять группы в большом наборе элементов, что, в свою очередь, обеспечивает более эффективное определение и манипулирование большими группами элементов.

Вторым предпочтением является то, что свернутое состояние все еще обеспечивает
30 прямой доступ одним нажатием к нескольким видимым элементам. Полагая, что видимые элементы были выбраны по их «важности» для пользователя (например, самые последние, те, к которым наиболее часто обращались в прошлом, и т. д.), видимые элементы - это те, которые наиболее вероятно ищет пользователь. Например, чтобы распечатать изображение, недавно кому-либо отправленное, пользователь
35 может прокрутить к группе Pictures и файл должен быть точно на вершине списка (как последний, к которому обращались). Можно сравнить это с программой просмотра текущих элементов - если свернутое положение изображения показано в пиктограмме папки, пользователю все еще нужно открыть папку, чтобы обратиться к файлу.
40 Наконец, сжатый вид является промежуточным между свернутым и развернутым состояниями: он пытается сбалансировать как целое визуализацию групп и их манипулирование с доступом к индивидуальным элементам.

Так как полусвернутый вид обеспечивает удобный способ доступа к выбранным
45 элементам из группы (без необходимости обращаться ко всем элементам в группе), пользователям может быть предоставлен контроль над тем, какие элементы показаны в полусвернутом виде и сколько элементов появилось. В одном подходе элементы могут быть отсортированы по predetermined критериям, и эти элементы показаны на вершине отсортированного списка. Пользователь может изменить
50 критерии сортировки и число показываемых элементов. Например, удобным и полезным способом сортировки документов является сортировка по дате последних изменений. Полусвернутый вид может показать первые n документов, последних, к которым обращались, из списка по умолчанию и может иметь кнопку, чтобы

показать следующие п. Альтернативой является иметь кнопку, чтобы показывать сохраненные документы за сегодня, вчера, прошлую неделю, прошлый месяц и т. д. Обычно во всех этих случаях показанный порядок элементов один и тот же, используемый, чтобы ограничить видимые элементы. Тем не менее, другим подходом является упорядочивать элементы способом, наиболее удобным для пользователя и не обязательно по тому же критерию, который используется для выбора элементов. Например, людей обычно лучше всего сортировать по алфавиту, даже если порядок выбора - по «важности».

Элементы в сжатой группе могут быть отображены как полусвернутый список. Полусвернутый список может быть выборочно развернут, чтобы показать больше элементов. (Альтернативно может быть развернута целая группа, чтобы показать все элементы.) Вид полусвернутого списка может быть использован для любого типа элементов и когда порядок сортировки отличается от порядка выбора. (Можно также использовать полуразвернутый вид списка, если порядок сортировки и порядок выбора один и тот же.) Примером является список любимых песен, отсортированный по алфавиту. Пользователь может развернуть части списка, чтобы показать менее популярные песни, но следующие видимые песни будут выбраны по их популярности.

При создании иерархии свойств кластеры высшего уровня обычно включают содержание всех вложенных кластеров. Например, кластер Documents включает все Word Documents, Excel Worksheets и так далее. Подобным образом элементы за 2003 год включают элементы за индивидуальные месяцы, которые, в свою очередь, включают индивидуальные дни. Любой контейнер (кластер или папка) может рассматриваться как самостоятельный элемент, чтобы управлять им как отдельным объектом, или просто как группа элементов, используемых, чтобы упорядочить вид.

Первичной функцией программы визуализации элементов является давать возможность легко находить требуемый элемент(ы). Тем не менее, обход кластеров свойств является только одним из способов. Функциональные возможности быстрого просмотра могут быть сильно расширены разрешением горизонтальных поисков, которые приходят к некоторым соответствующим элементам быстрее, чем при практической обработке иерархии свойств. Кроме того, программа визуализации должна позволять упорядочивание элементов любым способом, определяемым пользователем. При поиске элементов пользователи часто работают со связанными вместе элементами. Например, точная дата последнего редактирования документа может быть не известна, но пользователь может помнить, что это было сразу после важной встречи. Саму встречу можно легко найти, с точки которой наиболее подходящим запросом является «показать все документы, начиная с этой даты».

Со ссылкой на Фиг.11 примерная среда 1110 для реализации различных объектов изобретения включает компьютер 1112. Компьютер 1112 включает устройство 1114 обработки данных, системную память 1116 и системную шину 1118. Системная шина 1118 соединяет компоненты системы, включающие, но не ограничивающиеся, системную память 1116 и устройство 1114 обработки данных. Устройство 1114 обработки данных может быть любым из различных возможных процессоров. Сдвоенные микропроцессоры и другие мультипроцессоры также могут быть применены как устройство 1114 обработки данных.

Системная шина 1118 может быть любой из нескольких типов структур шин, включающих шину памяти или контроллер памяти, периферийную шину или внешнюю шину и/или локальную шину, использующую любое разнообразие шинных архитектур, причем архитектуры включают, но не ограничиваются, 16-битовую шину,

архитектуру, соответствующую промышленному стандарту (ISA), микроканальную архитектуру (MSA), расширенную стандартную архитектуру для промышленного применения (EISA), интеллектуальный интерфейс накопителей (IDE), локальную шину VESA (VLB), шину соединения периферийных компонентов (PCI), универсальную последовательную шину (USB), расширенный графический порт (AGP), Международную ассоциацию производителей плат памяти для персональных компьютеров (PCMCIA) и интерфейс малых компьютерных систем (SCSI).

Системная память 1116 включает энергозависимую память 1120 и энергонезависимую память 1122. Базовая система ввода/вывода (BIOS), содержащая основные рутинные операции, чтобы передавать информацию между элементами в компьютере 1112, такую как во время старта, сохранена в энергонезависимой памяти 1122. В качестве иллюстрации, но не ограничиваясь ею, энергонезависимая память 1122 может включать постоянное запоминающее устройство (ROM), программируемое ROM (PROM), электрически программируемое ROM (EPROM), электрически-стираемое программируемое ROM (EEPROM) или флэш-память. Энергозависимая память 1120 включает оперативное запоминающее устройство (RAM), которое действует как внешний кэш. В качестве иллюстрации, но не ограничиваясь ею, RAM возможно во многих формах, таких как синхронное RAM (SRAM), динамическое RAM (DRAM), синхронное DRAM (SDRAM), SDRAM с удвоенной скоростью данных (DDR SDRAM), усовершенствованное SDRAM (ESDRAM), DRAM с синхронизированной связью (SLDRAM) и RAM с шиной прямого резидентного доступа (DRRAM).

Компьютер 1112 также включает съемные/несъемные, энергозависимые/энергонезависимые компьютерные носители памяти. Фиг.11 иллюстрирует, например, запоминающее устройство 1124 на дисках. Запоминающее устройство 1124 на диске включает, но не ограничивается, такие устройства, как магнитный накопитель на дисках, гибкий накопитель на дисках, накопитель на ленте, Jaz-накопители, Zip-накопители, LS-100-накопители, плату флэш-памяти или memory stick. Дополнительно, запоминающее устройство 1124 на дисках может включать носитель памяти отдельно или в сочетании с другим носителем памяти, включающим, но не ограничивающимся, оптический накопитель на дисках, такой как компакт-дисковое ROM-устройство (CD-ROM), CD записываемый накопитель (CD-R-Накопитель), CD перезаписываемый накопитель (CD-RW-Накопитель) или CD-накопитель на универсальном цифровом диске (DVD-ROM). Чтобы облегчить связь запоминающих устройств 1124 на дисках для системной шины 1118, обычно используется съемный или несъемный интерфейс, такой как интерфейс 1126.

Должно быть оценено, что Фиг.11 описывает программное обеспечение, которое действует как посредник между пользователями и основными компьютерными ресурсами, описанными в подходящей операционной среде 1110. Такое программное обеспечение включает операционную систему 1128. Операционная система 1128, которая может быть сохранена на запоминающем устройстве 1124 на дисках, действует, чтобы контролировать и распределять ресурсы компьютерной системы 1112. Системные приложения 1130 используют в своих интересах управление ресурсами операционной системы 1128 через программные модули 1132 и программные данные 1134, сохраненные или в системной памяти 1116, или в запоминающем устройстве 1124 на дисках. Должно быть оценено, что настоящее изобретение может быть реализовано с различными операционными системами или сочетаниями операционных систем.

Пользователь вводит команды или информацию в компьютер 1112 через устройство(а) 1136 ввода. Устройства 1136 ввода включают, но не ограничиваются, указательные устройства, такие как мышь, шаровой указатель, перо, сенсорную клавиатуру, клавиатуру, микрофон, джойстик, игровую клавиатуру, спутниковую тарелку, сканнер, карту TV-тюнера, цифровую камеру, цифровую видеокамеру, web-камеру и тому подобное. Эти и другие устройства ввода связаны с устройством 1114 обработки данных через системную шину 1118 via порт(ы) 1138 интерфейса. Порт(ы) 1138 интерфейса включают, например, последовательный порт, параллельный порт, игровой порт и универсальную последовательную шину (USB). Устройство(а) 1140 вывода использует некоторые из этих же типов портов, как и устройство(а) 1136 ввода. Таким образом, например, порт USB может быть использован, чтобы обеспечить ввод в компьютер 1112 и вывод информации из компьютера 1112 на устройство 1140 вывода. Адаптер 1142 вывода обеспечивается, чтобы проиллюстрировать, что существует несколько устройств 1140 вывода, таких как мониторы, динамики и принтеры, среди остальных устройств 1140 вывода, которые требуют специальные адаптеры. Адаптеры 1142 вывода включают, в качестве примера, но не ограничиваясь им, видео и звуковые карты, которые обеспечивают средство связи между устройством 1140 вывода и системной шиной 1118. Следует заметить, что другие устройства и/или системы устройств, такие как удаленный(е) компьютер(ы) 1144, обеспечивают как возможность ввода, так и возможность вывода.

Компьютер 1112 может действовать в среде с сетевой структурой, используя логические связи с одним или более удаленными компьютерами, такими как удаленный(е) компьютер(ы) 1144. Удаленный(е) компьютер(ы) 1144 может быть персональным компьютером, сервером, маршрутизатором, сетевым PC, рабочей станцией, аппаратом, основанным на микропроцессоре, одноранговым устройством или другим обычным главным узлом и тому подобное и обычно содержит много или все элементы, описанные в отношении компьютера 1112. Для краткости только устройство 1146 хранения памяти проиллюстрировано с удаленным(и) компьютером(ами) 1144. Удаленный(е) компьютер(ы) 1144 логически связан с компьютером 1112 через сетевой интерфейс 1148 и затем физически связан via соединение 1150 связи. Сетевой интерфейс 1148 окружает сети связи, такие как локальные сети (LAN) и глобальные сети (WAN). Технологии LAN включают распределенный интерфейс передачи данных по волоконно-оптическим каналам (FDDI), распределенный проводной интерфейс передачи данных (CDDI), Ethernet/IEEE 1102.3, Token Ring/IEEE 1102.5 и тому подобное. Технологии WAN включают, но не ограничиваются, двухточечную связь, сети с коммутацией каналов, такие как цифровые сети связи с комплексными услугами (ISDN) и их варианты, сети с коммутацией пакетов и цифровые абонентские линии (DSL).

Соединение(я) 1150 связи ссылается на аппаратное/программное обеспечение, применяемое, чтобы соединить сетевой интерфейс 1148 с шиной 1118. Хотя соединение 1150 связи показано для иллюстративной ясности внутри компьютера 1112, оно может быть также вне компьютера 1112. Аппаратное/программное обеспечение, необходимое для соединения с сетевым интерфейсом 1148, включает, только с целью примера, внутренние и внешние технологии, такие как модемы, включающие обычные телефонные модемы, кабельные модемы и DSL-модемы, ISDN-адаптеры и карты Ethernet.

Фиг.12 является схематической блок-схемой примерной вычислительной среды

1200, с которой настоящее изобретение может взаимодействовать. Система 1200 включает один или более клиент(ов) 1210. Клиент(ы) 1210 может быть аппаратным/программным обеспечением (например, потоками, процессами, вычислительными устройствами). Система 1200 также включает один или более сервер(ов) 1230. Сервер(ы) 1230 может быть также аппаратным/программным обеспечением (например, потоками, процессами, вычислительными устройствами). Серверы 1230 могут вмещать потоки, чтобы выполнять преобразования, применяя настоящее изобретение, например. Одна возможная связь между клиентом 1210 и сервером 1230 может быть в форме пакета данных, приспособленного к передаче между двумя или более вычислительными процессами. Система 1200 включает структуру 1250 связи, которая может быть применена, чтобы облегчить связь между клиентом(ами) 1210 и сервером(ами) 1230. Клиент(ы) 1210 действительно связан с одним или более информационным(и) складом(ами) 1260 клиента, которые могут быть применены, чтобы хранить информацию, локальную для клиента(ов) 1210. Подобным образом сервер(ы) 1230 действительно связан с одним или более информационным(и) складом(ами) 1240 сервера, которые могут быть применены, чтобы хранить информацию, локальную для сервера(ов) 1230.

Описанное выше включает примеры настоящего изобретения. Конечно, невозможно описать каждое мыслимое сочетание компонент или методологий для целей описания настоящего изобретения, но специалист в данной области техники может признать, что возможны многие дополнительные сочетания и изменения настоящего изобретения. Следовательно, настоящее изобретение подразумевает охват всех таких изменений, модификаций и вариаций, которые являются частью сущности и объема изобретения, определяемого приложенной формулой изобретения. Кроме того, в пределах, в которых термин «включает» используется или в подробном описании, или в формуле изобретения, такой термин подразумевает включение в себя, так же как для термина «содержащий» «содержащий» интерпретируется как переходное слово в формуле изобретения.

Формула изобретения

1. Компьютеризированный интерфейс для представления данных, содержащий:
 - анализатор свойств, который определяет распределение элементов и формирует множество кластеров первого уровня, основываясь частично на первом свойстве из множества свойств, и автоматически определяет соответствующее распределение элементов для каждого другого свойства из множества свойств, вычисляет оценку кластеризации для каждого другого свойства, основываясь на соответствующем распределении элементов, и выбирает одно другое свойство, имеющее самую высокую оценку кластеризации; и
 - организатор, который автоматически формирует множество новых кластеров, основываясь частично на одном другом свойстве, и представляет множество новых кластеров.
2. Интерфейс по п.1, в котором множество свойств связаны с множеством элементов, множество элементов хранится по меньшей мере в одной из локальной или удаленной ячейки, или их комбинации, запоминающего устройства.
3. Интерфейс по п.2, в котором множество элементов включает в себя по меньшей мере один из: документ, файл, папку, изображение, аудио файл, видео файл, код, сообщение или компьютерное представление внешних объектов, включающих в себя людей или местоположения, или их комбинацию.

4. Интерфейс по п.1, в котором множество свойств содержит тип элемента, дату или время создания элемента, людей, связанных с элементом, месторасположение элемента, категорию элемента, или систему, приложение, администратора или пользователя, определяющих свойство элемента, или их комбинацию.

5. Интерфейс по п.1, в котором первым свойством элемента является тип элемента.

6. Интерфейс по п.1, в котором оценка кластеризации вычисляется следующим уравнением: оценка кластеризации =

$n_{\text{элементов.под.кластер1}} * n_{\text{элементов.под.кластер2}} * \dots$, для всех кластеров,

связанных с конкретным свойством множества свойств, где $n_{\text{элементов}}$ является количеством элементов, связанных с соответствующим кластером.

7. Интерфейс по п.1, в котором оценка кластеризации вычисляется следующим уравнением: оценка = $(N_{\text{суммарное}})! / ((n_{\text{элементов.под.кластер1}})! * (n_{\text{элементов.под.кластер2}})! * \dots)$, для всех кластеров, связанных с конкретным свойством множества

свойств, где $N_{\text{суммарное}}$ является общим количеством элементов всех кластеров и $n_{\text{элементов}}$ является количеством элементов, связанных с соответствующим

кластером.

8. Интерфейс по п.1, который также содержит интерфейс пользователя, чтобы по меньшей мере отображать результаты кластеров, получать выборку запросов, получать информацию о свойствах, или отображать информацию, относящуюся к элементу данных в кластере, или их комбинацию.

9. Считываемая компьютером среда, имеющая хранящиеся на ней считываемые компьютером инструкции для реализации анализатора свойств и организатора кластеров по п.1.

10. Графический интерфейс пользователя, содержащий:

один или более элементов данных и соответствующих свойств, хранящихся в базе данных;

один или более объектов отображения, созданных для каждого из элементов данных;

компоненту ввода для выбора элементов данных и соответствующих свойств;

анализатор свойств, который формирует множество кластеров, основываясь

частично на первом свойстве из множества свойств, и автоматически вычисляет

оценку кластеризации для каждого другого свойства, и выбирает одно другое

свойство из множества свойств, имеющее самую высокую оценку кластеризации, и

определяет распределение элементов, основываясь частично на одном другом свойстве;

компоненту организатора, который создает новые кластеры и хранит объекты отображения, связанные с элементами данных в соответствующем новом кластере, основываясь на одном другом свойстве; и

компоненту отображения, чтобы представлять объекты отображения и новые кластеры, частично основываясь на автоматизированном анализе свойств.

11. Интерфейс по п.10, который также содержит управляющие элементы для взаимодействия со свойствами.

12. Интерфейс по п.11, в котором свойства применяются для вложенного запроса результатов.

13. Интерфейс по п.11, в котором свойства включают в себя по меньшей мере одно из: тип, месторасположение, категория, личность, дата, время и параметр, определяемый пользователем, или их комбинацию.

14. Интерфейс по п.11, который также содержит компоненту, чтобы косвенным

образом обучаться на действиях пользователя.

15. Интерфейс по п.11, который также содержит по меньшей мере один полусвернутый список или группу.

5 16. Интерфейс по п.15, который также содержит управляющие элементы для расширения списка или группы.

17. Интерфейс по п.16, в котором по меньшей мере один большой кластер свойств представлен в сжатом виде, использующим полусвернутый список.

10 18. Компьютеризированный интерфейс для оптимизации извлечения и отображения информации, содержащий:

анализатор свойств, который определяет распределение элементов и формирует множество кластеров, основываясь частично на первом свойстве из множества свойств, автоматически вычисляет оценку кластеризации для каждого другого свойства и выбирает по меньшей мере одно другое свойство из множества свойств, основываясь на оценке кластеризации, связанной с по меньшей мере одним другим свойством, и определяет распределение множества элементов, основываясь частично на по меньшей мере одном другом свойстве; и

15 организатор, который автоматически формирует множество оптимизированных кластеров и распределяет множество элементов в множестве оптимизированных кластеров, основываясь частично на по меньшей мере одном другом свойстве, и представляет множество оптимизированных кластеров.

20 19. Интерфейс по п.18, в котором множество элементов хранится по меньшей мере в одной из локальной или удаленной ячеек, или их комбинации, запоминающего устройства.

25 20. Интерфейс п.19, в котором множество элементов включает в себя по меньшей мере два из: документ, файл, папку, изображение, аудио файл, видео файл, код, сообщение или компьютерное представление внешних объектов, включающих в себя людей или местоположения, или их комбинацию.

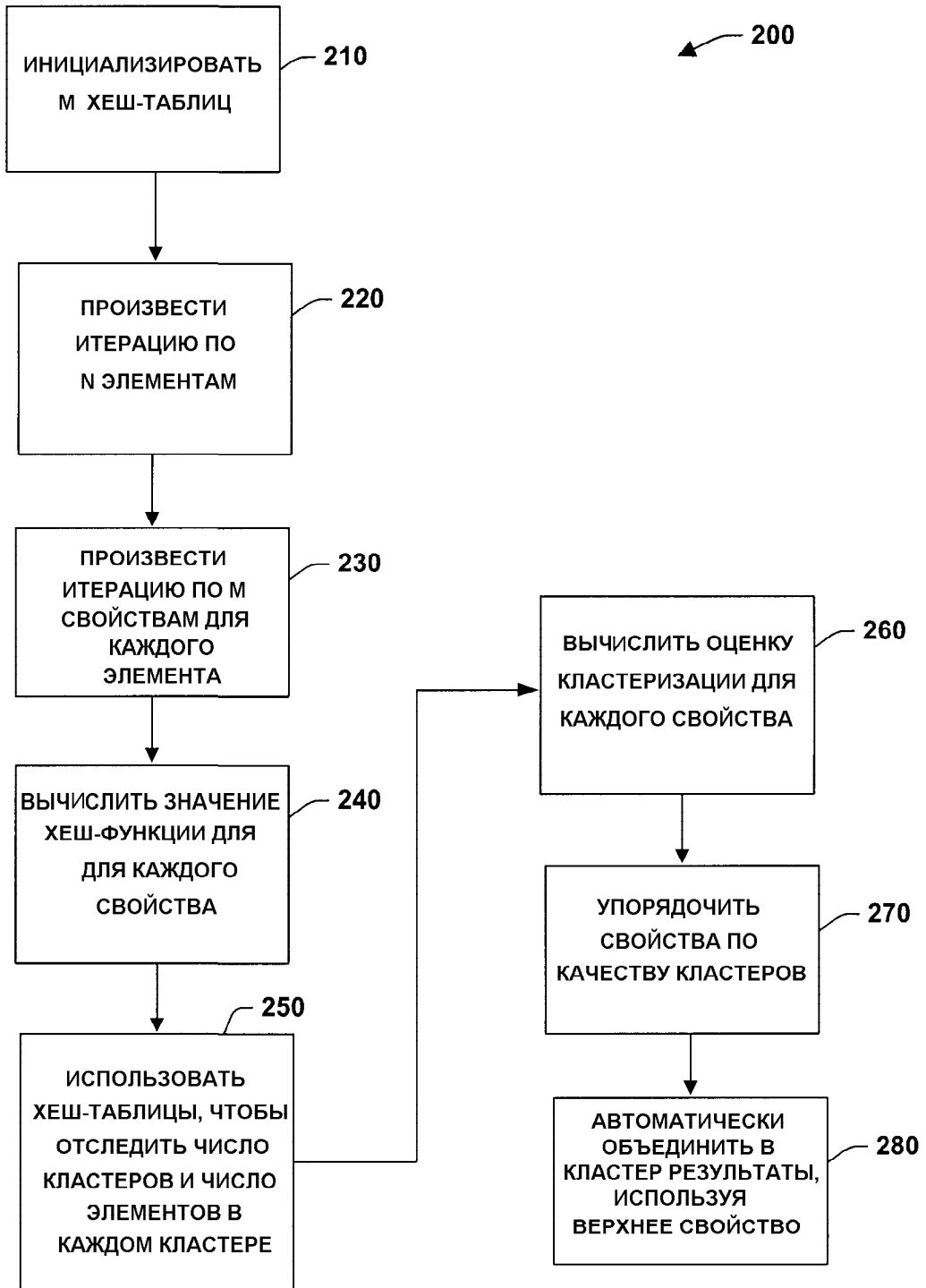
30 21. Интерфейс п.18, в котором множество свойств содержит тип элемента, дату или время создания элемента, людей, связанных с элементом, месторасположение элемента, категорию элемента, или систему, приложение, администратора или пользователя, определяющих свойство элемента, или их комбинацию.

35 22. Интерфейс по п.21, в котором множество свойств также содержит метаданные.

40 23. Интерфейс п.18, в котором анализатор свойств выбирает по меньшей мере одно другое свойство, основываясь на самой высокой оценке кластеризации, оценка кластеризации равна результату произведения количества элементов в соответствующем кластере на количество элементов в каждом другом соответствующем кластере, или равен общему количеству элементов, разделенному на результат от факториала от количества элементов в соответствующем кластере, умноженному на факториал от количества элементов в каждом другом соответствующем кластере.

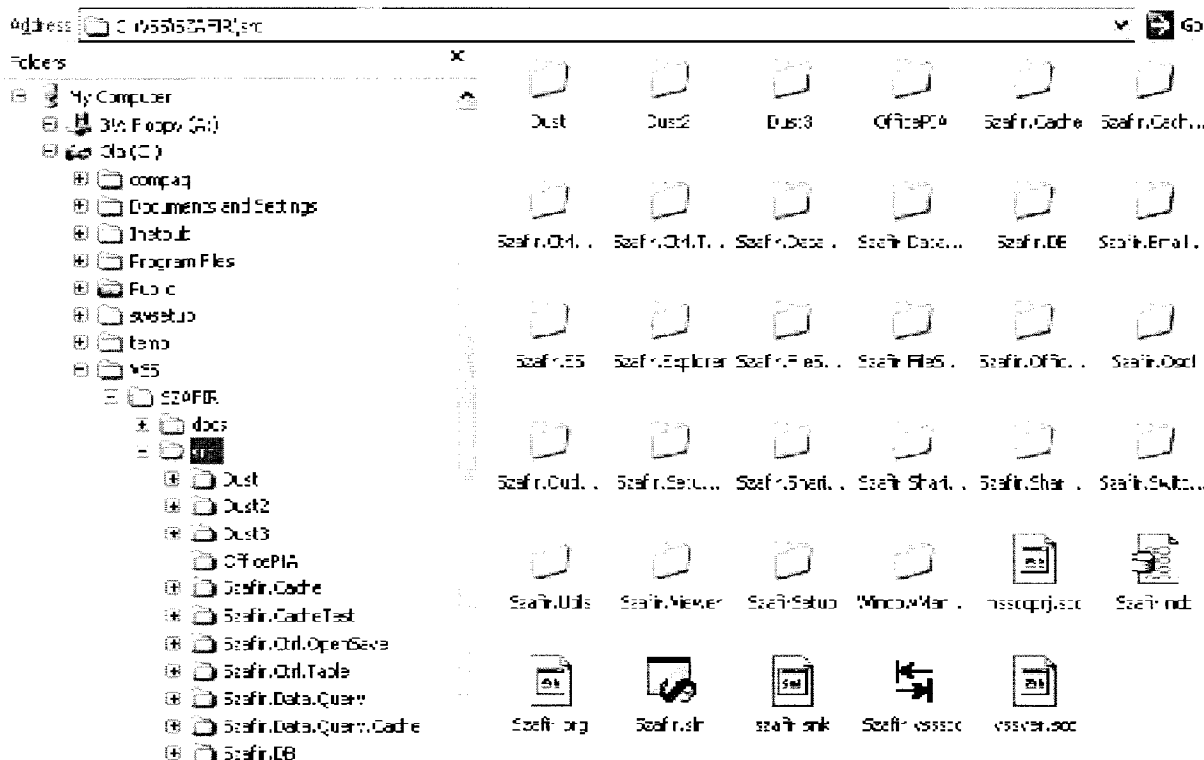
45 24. Интерфейс п.18, в котором по меньшей мере множество оптимизированных кластеров или множество элементов, или их комбинация, представлены в полусвернутом виде, основываясь на критерии значимости.

50



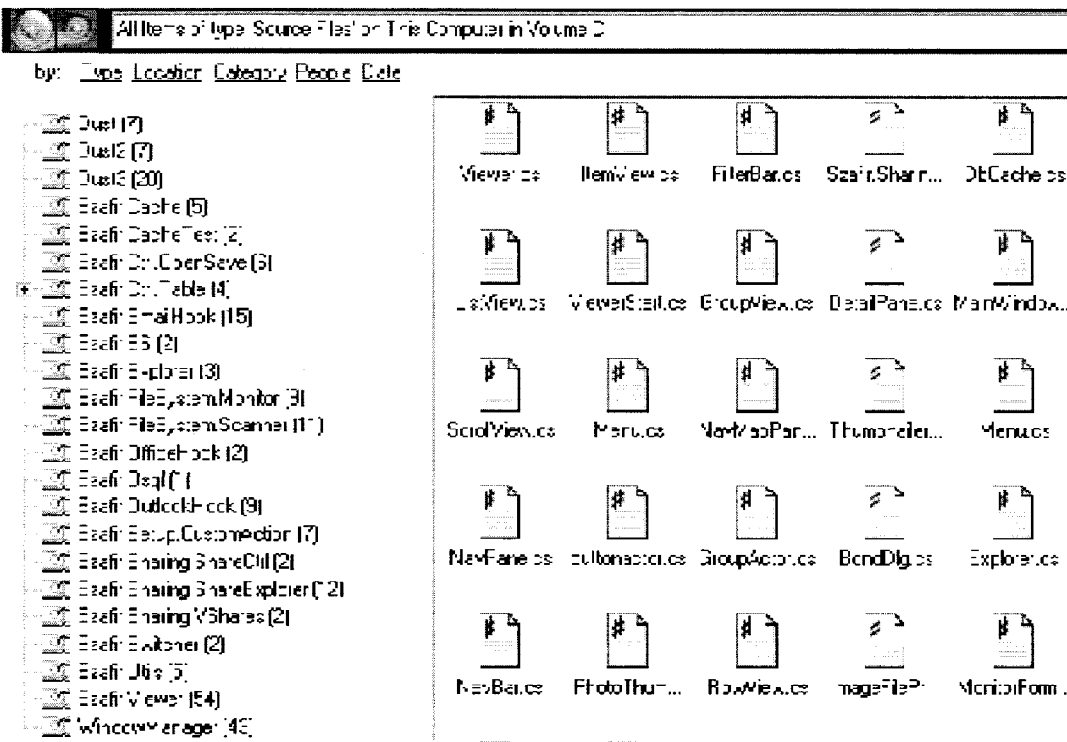
ФИГ.2

300

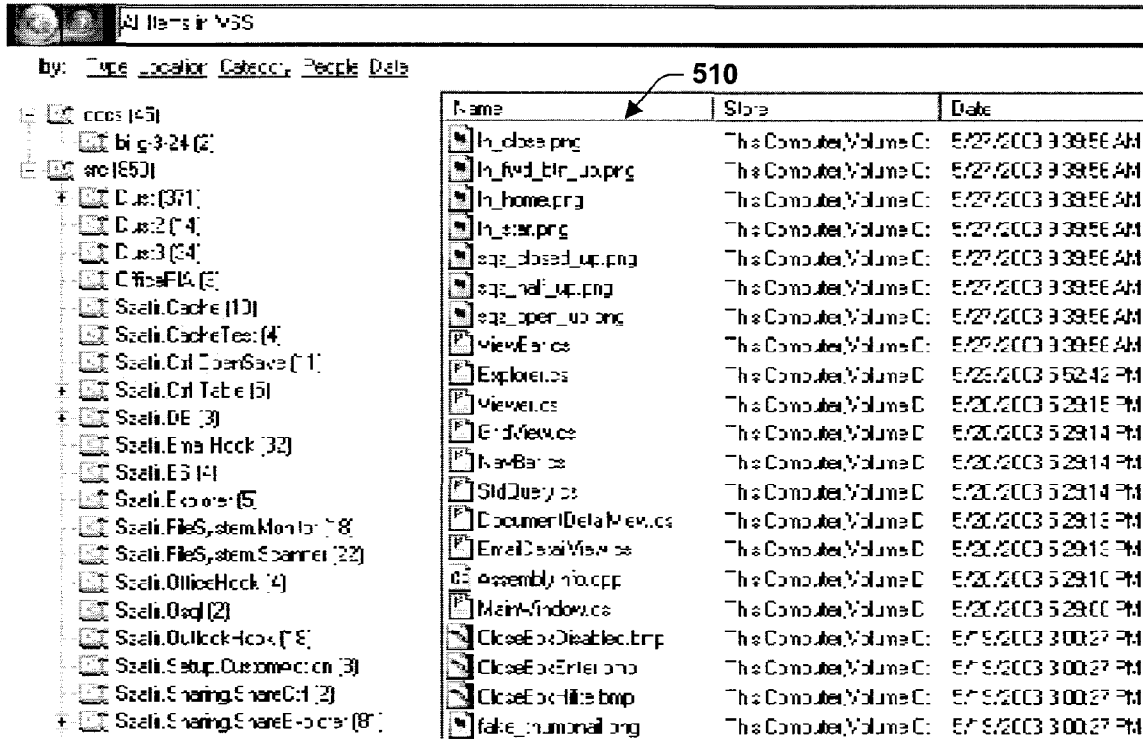


ФИГ.3

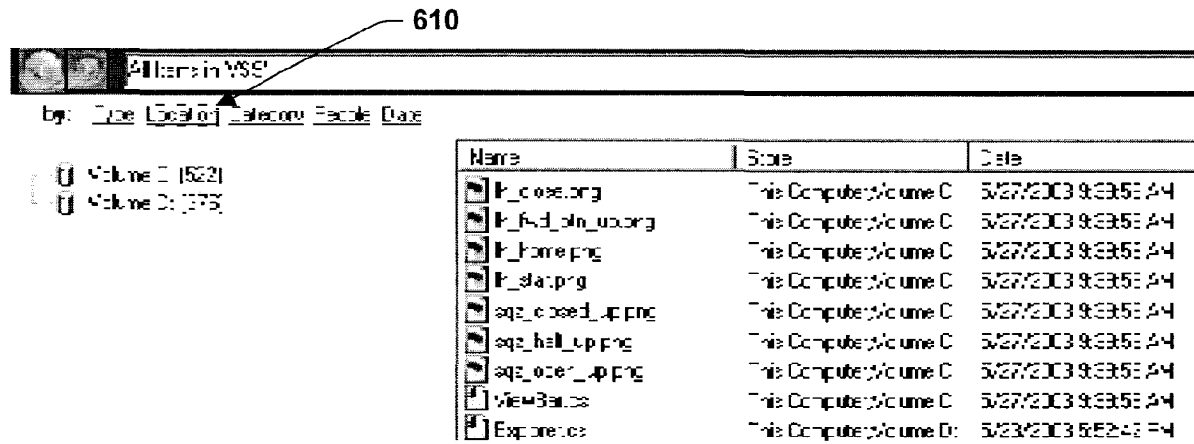
400



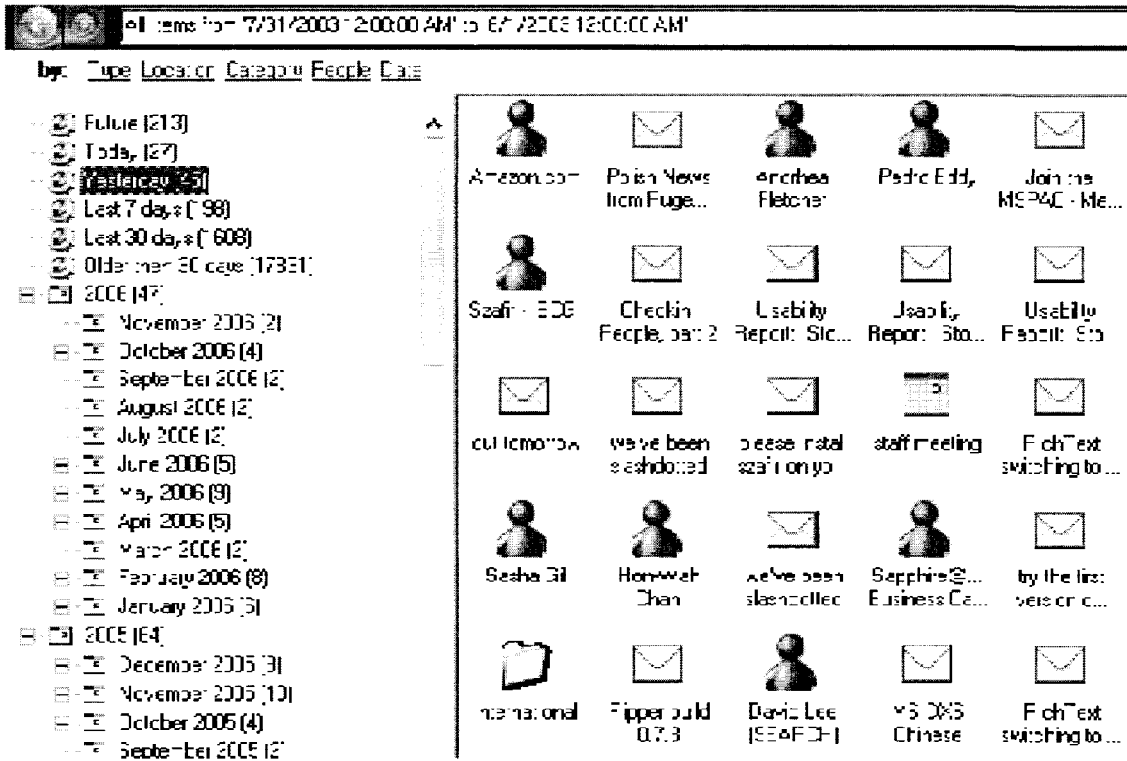
ФИГ.4



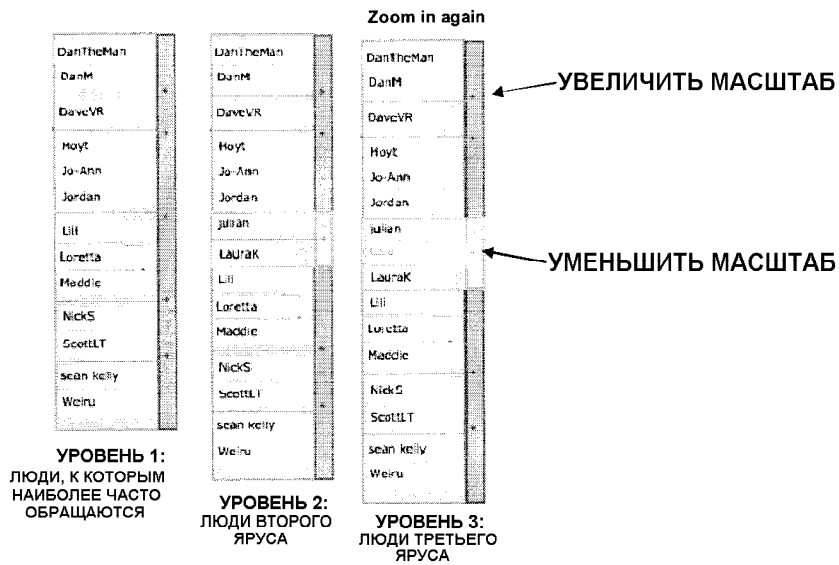
ФИГ.5



ФИГ.6

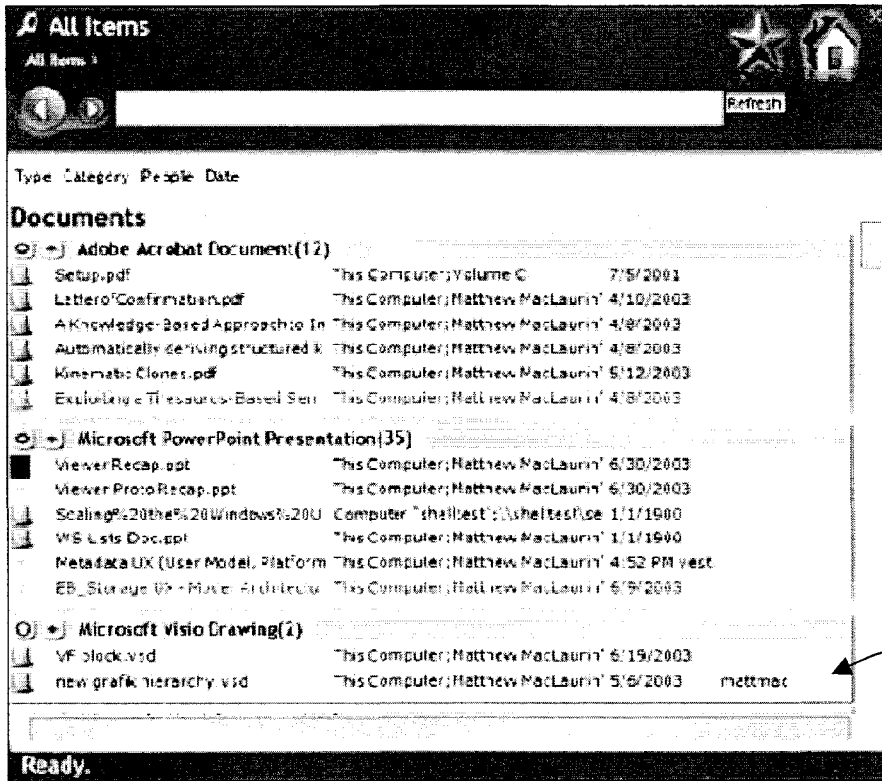


ФИГ.7



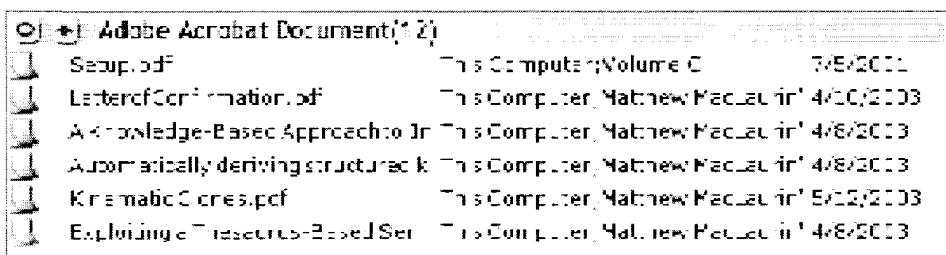
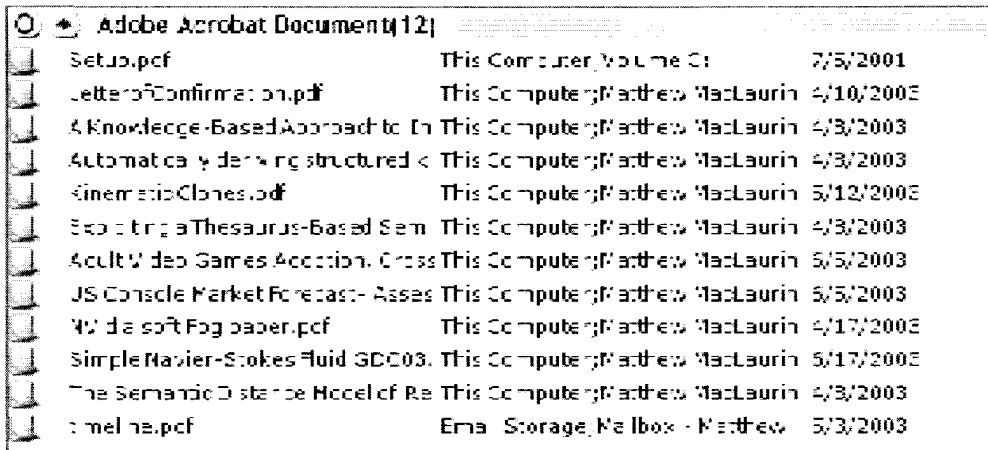
ФИГ.8

900

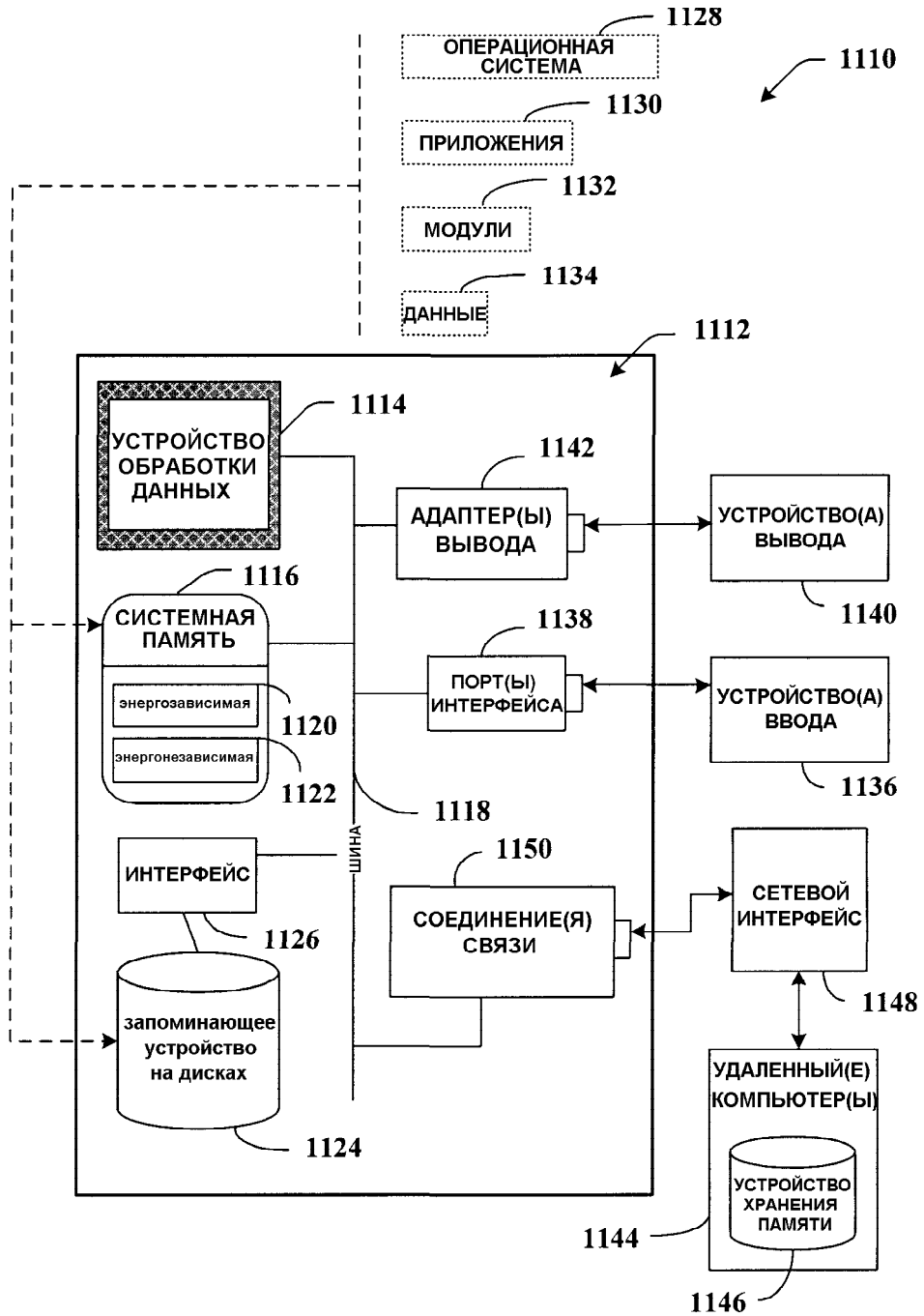


ФИГ.9

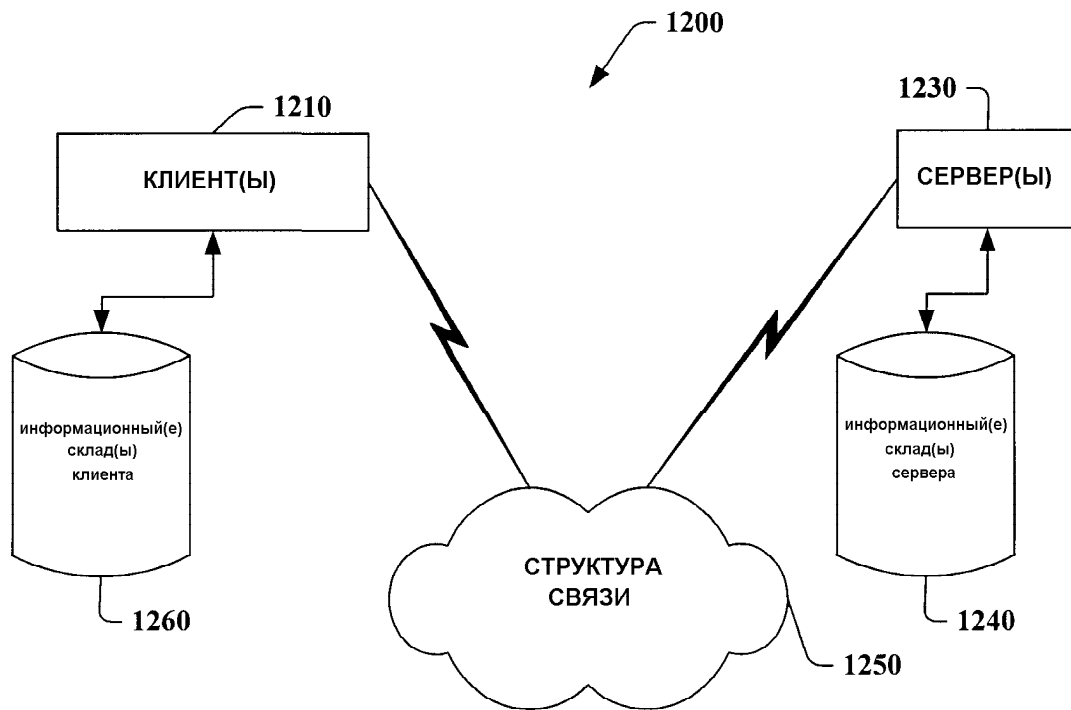
1000



ФИГ.10



ФИГ.11



ФИГ.12