



ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ,
ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21), (22) Заявка: **2007118486/09**, **18.11.2005**

(24) Дата начала отсчета срока действия патента:
18.11.2005

(30) Конвенционный приоритет:
18.11.2004 KR 10-2004-0094795

(43) Дата публикации заявки: **27.11.2008**

(45) Опубликовано: **10.01.2010** Бюл. № 1

(56) Список документов, цитированных в отчете о
поиске: **RU 2217875 C2**, **27.11.2003**. **US 6754703**,
22.06.2004. **KR 20010090254**, **18.10.2001**. **WO**
0124444 A1, **05.04.2001**. **WO 02077830 A1**,
03.10.2002.

(85) Дата перевода заявки РСТ на национальную
фазу: **17.05.2007**

(86) Заявка РСТ:
KR 2005/003919 (18.11.2005)

(87) Публикация РСТ:
WO 2006/054877 (26.05.2006)

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):

СОНГ Бонг-Гиу (KR),
ЧОИ Сеунг-Пил (KR),
ДЗОЕ Вон-Чанг (KR)

(73) Патентообладатель(и):

САМСУНГ ЭЛЕКТРОНИКС КО., ЛТД.
(KR)

(54) УСТРОЙСТВО И СПОСОБ УПРАВЛЕНИЯ СЕТЬЮ НА ОСНОВЕ ПРОСТОГО ПРОТОКОЛА УПРАВЛЕНИЯ СЕТЬЮ (SNMP)

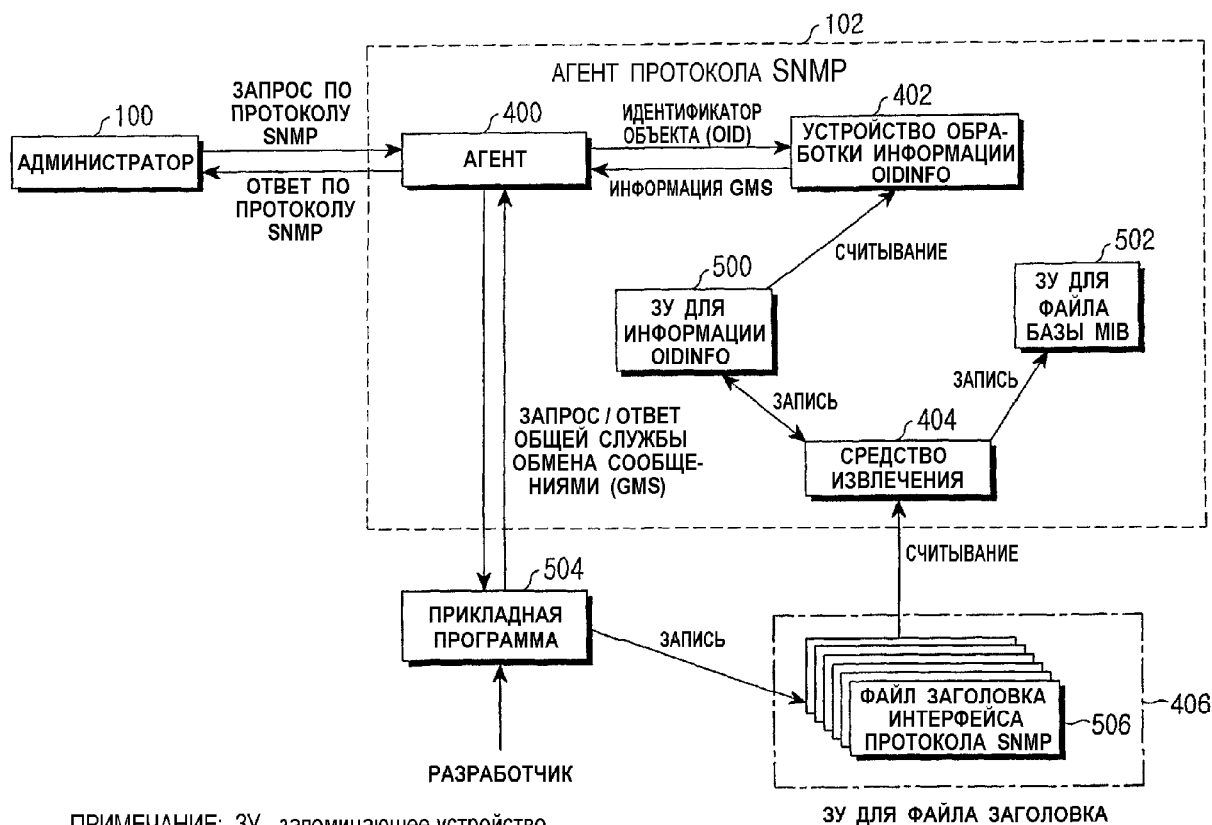
(57) Реферат:

Изобретение относится к сетям передачи данных. Технический результат заключается в усовершенствовании механизма управления сетями. Предложены устройство и способ для административного управления устройством связи с использованием простого протокола управления сетью (SNMP). При создании разработчиком заголовочного файла интерфейса протокола SNMP посредством прикладной программы во время компиляции

средство извлечения на основании заголовочного файла интерфейса осуществляет генерацию файла базы управляющей информации (MIB) и информации об идентификаторе объекта (OIDInfo). Когда во время выполнения программы администратор выдает запрос согласно протоколу SNMP, то агент передает информацию OIDInfo, содержащуюся в сообщении с запросом согласно протоколу SNMP, в устройство обработки информации OIDInfo. Устройство

обработки информации OIDInfo обращается к запоминающему устройству для информации OIDInfo и передает агенту информацию общей службы обмена сообщениями (GMS). Затем выполняют процесс передачи запроса/ответа общей службы обмена

сообщениями (GMS) между агентом и прикладной программой на основании информации общей службы обмена сообщениями (GMS). 3 н. и 7 з.п.ф-лы, 18 ил., 24 табл.



ПРИМЕЧАНИЕ: ЗУ - запоминающее устройство

Фиг. 5



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY,
PATENTS AND TRADEMARKS

(12) ABSTRACT OF INVENTION(21), (22) Application: **2007118486/09, 18.11.2005**(24) Effective date for property rights:
18.11.2005(30) Priority:
18.11.2004 KR 10-2004-0094795(43) Application published: **27.11.2008**(45) Date of publication: **10.01.2010 Bull. 1**(85) Commencement of national phase: **17.05.2007**(86) PCT application:
KR 2005/003919 (18.11.2005)(87) PCT publication:
WO 2006/054877 (26.05.2006)

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):

**SONG Bong-Giu (KR),
ChOI Seung-Pil (KR),
DZOE Von-Chang (KR)**

(73) Proprietor(s):

SAMSUNG EhLEKTRONIKS KO., LTD. (KR)

(54) DEVICE AND METHOD OF MANAGING NETWORK BASED ON SIMPLE NETWORK MANAGEMENT PROTOCOL (SNMP)

(57) Abstract:

FIELD: physics; communications.

SUBSTANCE: invention relates to data transmission networks. A device and a method are proposed for administration management of a communication device using a simple network management protocol (SNMP). When a developer creates a header file of the SNMP interface through an application program during compilation apparatus for extraction based on the interface header file generates a management information base (MIB) file and object identifier information (OIDInfo). If the administrator sends a request in accordance with SNMP during execution of the program, the agent then sends OIDInfo contained in the message with the request in accordance with SNMP to an OIDInfo processing device. The OIDInfo processing device accesses a memory unit for OIDInfo and sends the agent global messaging service information (GMS).

Request/response of the global messaging service (GMS) is then transmitted between the agent and the application program based on the global messaging service information (GMS).

EFFECT: improved mechanism of managing networks.

10 cl, 19 dwg, 24 tbl



Область техники, к которой относится изобретение

Настоящее изобретение относится, в общем случае, к устройству и к способу управления сетевым устройством. В частности, настоящее изобретение относится к устройству административного управления сетью и к способу административного управления устройством связи с использованием простого протокола управления сетью (SNMP).

Уровень техники

Комплексное управление сетью является затруднительным вследствие быстрого роста сетей за несколько последних лет и появления различных гетерогенных систем. Поскольку сети увеличивают масштаб, то управление сетевыми устройствами становится существенно важным во многих областях.

Следовательно, для администраторов сетей необходима сетевая инфраструктура для комплексного управления в различных сетевых средах. Вследствие этой потребности основной организацией по разработке стандартов для сети Интернет, то есть рабочей группой по инженерным проблемам сети Интернет (IETF), в качестве стандарта для управления сетевыми устройствами, основанного на протоколе сети Интернет, был утвержден Простой Протокол Управления Сетью (SNMP), соответствующий относительно простому протоколу.

В обычной системе, основанной на простом протоколе управления сетью (SNMP), систему управления именуют администратором, а целевой объект управления именуют агентом. Сеть передачи информации для административного управления, предназначенная для обеспечения связи администратора с агентом, основана на протоколе управления передачей/протоколе сети Интернет (TCP/IP) и при обмене информацией с использованием простого протокола управления сетью (SNMP) использует команду для извлечения управляющей информации, команду для последовательного извлечения управляющей информации, команду для изменения и записи управляющей информации и команду для оповещения о необычном функционировании на основании базы управляющей информации (MIB) путем передачи сообщений между администратором и агентом.

Агент простого протокола управления сетью (SNMP) представляет собой программный модуль, размещенный в устройстве, являющемся целевым объектом административного управления, и имеет информацию о базе управляющей информации (MIB). Эту информацию предоставляют администратору простого протокола управления сетью (SNMP) с использованием простого протокола управления сетью (SNMP).

Конкретную информацию, ресурсы и т.д., административное управление которыми должно быть осуществлено с использованием простого протокола управления сетью (SNMP), именуют объектами. Совокупность объектов именуют базой управляющей информации (MIB). Формат базы управляющей информации (MIB) определен как часть простого протокола управления сетью (SNMP), а объекты определены с использованием языка описания абстрактного синтаксиса версии 1 (ASN.1).

Агент простого протокола управления сетью (SNMP) осуществляет административное управление базой управляющей информации (MIB), сконфигурированной посредством параметров, относящихся к функции сетевого устройства. Администратор простого протокола управления сетью (SNMP) получает конкретное значение из баз управляющей информации (MIBs), обеспечиваемых агентами простого протокола управления сетью (SNMP), и распознает состояние

устройства или изменяет это значение.

Как описано выше, термин "операция обычного административного управления сетью с использованием простого протокола управления сетью (SNMP)" означает операцию получения конкретного значения из баз управляющей информации (MIB),
 5 обеспечиваемых устройствами, являющимися целевыми объектами административного управления, распознавания состояния устройства и изменения этого значения.

Согласно простому протоколу управления сетью (SNMP) может быть легко
 10 использован способ административного управления, и могут быть разработаны различные типы устройств, использующих протокол TCP/IP. Посредством различных документов, именуемых "запросами на комментарии" (RFCs), диапазон административного управления может быть легко назначен или расширен, и может
 15 быть обеспечено простое установление конфигурации протоколов. Среди многих протоколов административного управления простой протокол управления сетью (SNMP) находит широкое применение потому, что он может быть легко реализован.

На Фиг.1 проиллюстрирована обычная структура и операции управления,
 20 выполняемые между администратором 100 протокола SNMP и агентом 102 протокола SNMP.

Сначала приведено описание команд, передаваемых/принимаемых между администратором 100 протокола SNMP и агентом 102 протокола SNMP, которые
 25 показаны на Фиг.1.

GetRequest: обозначает сигнал запроса на считывание значения объекта;

GetNextRequest: обозначает сигнал запроса на считывание следующего значения
 объекта, являющегося следующим за текущим значением объекта;

GetResponse: обозначает сигнал ответа на запрос;

30 SetRequest: обозначает сигнал для записи значения объекта; и

Trap: обозначает сигнал для сообщения о необычной ситуации.

Администратор 100 протокола SNMP и агент 102 протокола SNMP, показанные на
 Фиг.1, могут осуществлять обмен информацией друг с другом с использованием
 описанных выше сообщений.

35 Теперь будет приведено описание обычного способа разработки и обычного способа взаимодействия с прикладной программой со ссылкой на Фиг.2 и Фиг.3.

На Фиг.2 показана схема последовательности операций, иллюстрирующая
 обычный способ разработки агента 102 протокола SNMP.

40 При операции 200 из Фиг.2 администратор сети определяет базу управляющей информации (MIB) для разработки агента 102 протокола SNMP и определяет структуру, используемую для интерфейса с соответствующей прикладной программой.

При операции 202 осуществляют генерацию файла базы управляющей
 информации (MIB) на основании базы управляющей информации (MIB), определенной
 45 при операции 200. При выполнении операции 204 администратор сети кодирует и компилирует сгенерированный файл базы управляющей информации (MIB),
 осуществляя, тем самым, генерацию агента 102 протокола SNMP.

При операции 206 администратор сети определяет, были ли внесены поправки в
 50 базу управляющей информации (MIB) или в интерфейс. Если в базу управляющей информации (MIB) или в интерфейс были внесены поправки, то администратор сети переходит к выполнению операции 208 для определения элементов административного управления.

При операции 210 администратор сети осуществляет генерацию файла базы управляющей информации (MIB) на основании элементов административного управления, определенных при выполнении операции 208. При операции 212 администратор сети производит повторное кодирование и повторную компиляцию

агента 102 протокола SNMP на основании сгенерированного файла базы управляющей информации (MIB).

На Фиг.3 проиллюстрирован обычный способ взаимодействия между администратором 100 и прикладными программами 304.

Для того чтобы обеспечить взаимодействие администратора 100 с прикладными программами 304, агент 102 должен знать информацию о структуре и об адресате. При использовании обычного инструментального средства для разработки агента 102 протокола SNMP объекты 302, соответствующие элементам административного управления, должны быть реализованы с использованием структуры, использованной

в прикладных программах 304. В обычном способе разработки агента 102 протокола SNMP базу управляющей информации (MIB) выполняют таким образом, чтобы обеспечить возможность отражения характеристик устройства. Содержимое спроектированной базы управляющей информации (MIB) определяет пределы разработки агента 102 протокола SNMP, роль прикладных программ 304 для выполнения функции административного управления в устройстве, способ взаимодействия между агентом 102 протокола SNMP и прикладными программами 304, и т.д.

Для практической разработки простого протокола управления сетью (SNMP) использовано обычное инструментальное средство. Разработка агента 102 протокола SNMP с использованием инструментального средства облегчена в том случае, когда агент 102 протокола SNMP имеет необходимые данные. Однако разработка агента 102 протокола SNMP является трудной и сложной потому, что объекты базы управляющей информации (MIB) имеют различные структуры, когда значение объекта базы управляющей информации (MIB) получено через интерфейс с прикладными программами 304.

Администратор 100 протокола SNMP осуществляет доступ к элементам административного управления, административное управление которыми в различных прикладных программах 304 в устройстве осуществляют посредством агента 102 протокола SNMP. Эти элементы административного управления выражены посредством базы управляющей информации (MIB) и являются различными в соответствии с характеристиками устройства. Эта база управляющей информации (MIB) не может быть точно определена в момент начальной разработки.

Поскольку разработку агента 102 протокола SNMP обычно производят на основании описанной выше базы управляющей информации (MIB), то для устройств должны быть разработаны различные агенты 102 протокола SNMP. Элементы, административное управления которыми должно быть осуществлено в идентичном устройстве, могут часто изменяться, добавляться или удаляться. Когда произошло изменение, то должна быть произведена корректировка агента 102 протокола SNMP.

Поскольку в описанной выше среде корректировку и повторную компиляцию агента 102 протокола SNMP необходимо производить при каждом изменении базы управляющей информации (MIB), при добавлении в нее данных или при удалении данных из нее, то для разработки требуется большое количество времени и усилий.

Следовательно, существует потребность в создании эффективной и действенной системы и способа для административного управления устройством связи с

использованием простого протокола управления сетью (SNMP).

СУЩНОСТЬ ИЗОБРЕТЕНИЯ

Предложенные варианты осуществления настоящего изобретения, по существу, решают вышеупомянутые и иные проблемы и обеспечивают устройство и способ управления сетью на основании простого протокола управления сетью (SNMP), которые обеспечивают генерацию файла базы управляющей информации (MIB) и файла информации об идентификаторе объекта в реальном масштабе времени в системе административного управления сетью.

Варианты осуществления настоящего изобретения обеспечивают устройство и способ административного управления на основании простого протокола управления сетью (SNMP) для управления работой устройств, которые являются пригодными для изменений, происходящих в сети, содержащей систему административного управления сетью.

Варианты осуществления настоящего изобретения обеспечивают устройство и способ административного управления на основании простого протокола управления сетью (SNMP) для управления работой устройств, которые содержат систему административного управления сетью согласно стандарту.

В соответствии с задачей настоящего изобретения в нем предложено устройство для генерации файла базы управляющей информации (MIB) для административного управления устройством связи с использованием простого протокола управления сетью (SNMP) во время компиляции и файла с информацией об идентификаторе объекта (OIDInfo), предназначенной для обмена ею между агентом простого протокола управления сетью (SNMP) и прикладной программой. Устройство содержит запоминающее устройство для заголовочного файла, предназначенное для запоминания заголовочного файла, созданного прикладной программой для административного управления устройством, работающим согласно простому протоколу управления сетью (SNMP), средство извлечения, предназначенное для считывания заголовочного файла из запоминающего устройства для заголовочного файла и создания файла базы управляющей информации (MIB) и файла OIDInfo для обмена сообщениями между агентом простого протокола управления сетью (SNMP) и прикладной программой, запоминающее устройство для файла базы управляющей информации (MIB), предназначенное для запоминания файла базы управляющей информации (MIB), созданного средством извлечения, и запоминающее устройство для файла OIDInfo, предназначенное для запоминания файла OIDInfo, созданного средством извлечения.

В соответствии с другой задачей настоящего изобретения предложен способ генерации файла базы управляющей информации (MIB) для административного управления устройством связи с использованием простого протокола управления сетью (SNMP) во время компиляции и файла с информацией об идентификаторе объекта (OIDInfo), предназначенной для обмена ею между агентом простого протокола управления сетью (SNMP) и прикладной программой. Способ содержит следующие операции: создают заголовочный файл интерфейса простого протокола управления сетью (SNMP) в прикладной программе, принимающей измененный элемент административного управления, считывают заголовочный файл интерфейса простого протокола управления сетью (SNMP) и осуществляют генерацию файла базы управляющей информации (MIB) и файла OIDInfo, и производят запоминание файла базы управляющей информации (MIB) и файла OIDInfo.

В соответствии с другой задачей настоящего изобретения предложен способ

выполнения во время работы операций приема от администратора сообщения с запросом согласно простому протоколу управления сетью (SNMP), содержащего элемент административного управления, для административного управления устройством связи с использованием протокола SNMP и выдачи ответа о результате

5 обработки элемента административного управления. Способ содержит следующие операции: в агенте принимают сообщение с запросом согласно протоколу SNMP для запроса на получение данных о результате обработки элемента административного управления, поступившее от администратора, производят передачу информации об

10 идентификаторе объекта (OIDInfo), содержащейся в сообщении с запросом согласно протоколу SNMP, из агента в устройство обработки информации OIDInfo, производят передачу информации, предназначенной для обмена ею между агентом и прикладной программой, в которой хранятся данные элемента административного управления, запрошенные администратором, из устройства обработки информации OIDInfo агенту

15 на основании информации OIDInfo, производят обмен информацией с прикладной программой в агенте на основании переданной информации, принятой из устройства обработки информации OIDInfo, и принимают результат обработки элемента административного управления, запрошенного администратором, из прикладной

20 программы, и передают принятый результат обработки элемента административного управления из агента администратору.

В соответствии с другой задачей настоящего изобретения предложено устройство для административного управления устройством связи с использованием простого протокола управления сетью (SNMP) для выполнения во время работы операций

25 приема от администратора сообщения с запросом согласно протоколу SNMP, содержащего элемент административного управления, для административного управления устройством связи с использованием протокола SNMP и выдачи ответа о результате обработки элемента административного управления. Устройство содержит

30 прикладную программу, предназначенную для выполнения операции выдачи запроса на получение элемента административного управления, средство обработки информации об идентификаторе объекта (OIDInfo), предназначенное для предоставления информации для обмена ею с прикладной программой при приеме

35 заранее заданной информации OIDInfo, поставленной в соответствие элементу административного управления, и агент, предназначенный для передачи информации OIDInfo, поставленной в соответствие элементу административного управления, в средство обработки информации OIDInfo при приеме от администратора сообщения с запросом согласно протоколу SNMP, для получения

40 информации, предназначенной для обмена ею с прикладной программой, для приема результата обработки элемента административного управления из прикладной программы и для передачи результата обработки администратору.

КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ

Вышеупомянутые и иные признаки и преимущества настоящего изобретения станут

45 более понятными из приведенного ниже подробного описания при его рассмотрении совместно с сопроводительными чертежами, на которых изображено следующее:

на Фиг.1 проиллюстрирована обычная структура и операции управления между администратором простого протокола управления сетью (SNMP) и агентом

50 протокола SNMP;

на Фиг.2 показана схема последовательности операций, на которой проиллюстрирован обычный способ разработки агента протокола SNMP из Фиг.1;

на Фиг.3 проиллюстрирован обычный способ взаимодействия между

администратором и прикладными программами;

на Фиг.4 изображена блок-схема, на которой проиллюстрирован пример агента протокола SNMP согласно варианту осуществления настоящего изобретения;

5 на Фиг.5 изображена блок-схема, на которой проиллюстрирован агент протокола SNMP, показанный на Фиг.4, и пример последовательности выполнения операций, включающей в себя время компиляции и время выполнения программы, согласно варианту осуществления настоящего изобретения;

10 на Фиг.6 изображена блок-схема, на которой проиллюстрирован пример операции компиляции агента протокола SNMP, показанного на Фиг.4, согласно варианту осуществления настоящего изобретения;

на Фиг.7 изображена схема последовательности операций, на которой проиллюстрирован пример функционирования агента протокола SNMP, показанного на Фиг.4, во время компиляции, согласно варианту осуществления настоящего изобретения;

на Фиг.8 проиллюстрирован пример способа выполнения ответа согласно простому протоколу управления сетью (SNMP) во время выполнения программы в агенте протокола SNMP, показанном на Фиг.4, с использованием информации об идентификаторе объекта, генерация которой произведена во время компиляции, показанной Фиг.6, согласно варианту осуществления настоящего изобретения;

на Фиг.9 изображена схема последовательности операций, на которой проиллюстрирован пример функционирования агента протокола SNMP, показанного на Фиг.4, во время выполнения программы, согласно варианту осуществления настоящего изобретения;

на Фиг.10 проиллюстрирован пример конфигурации протокольного блока данных (PDU) общей службы обмена сообщениями (GMS) для обмена информацией между агентом и прикладными программами согласно варианту осуществления настоящего изобретения;

на Фиг.11 проиллюстрирован пример процедуры передачи сообщения Get/GetNext/GetFirst из агента в прикладную программу и операции, связанной с ответом, согласно варианту осуществления настоящего изобретения;

на Фиг.12 проиллюстрирован пример процедуры передачи сообщения PreSet/Set из агента в прикладную программу и операции, связанной с ответом, согласно варианту осуществления настоящего изобретения;

на Фиг.13 проиллюстрирован пример процедуры передачи сообщения GetBulk из агента в прикладную программу и операции, связанной с ответом, согласно варианту осуществления настоящего изобретения;

на Фиг.14 проиллюстрирован пример процедуры передачи сообщения "уведомление" (Trap) из агента в прикладную программу и операции, связанной с ответом, согласно варианту осуществления настоящего изобретения;

на Фиг.15А схематично проиллюстрированы примеры элементов данных, содержащих файл с информацией об идентификаторе объекта, согласно варианту осуществления настоящего изобретения;

на Фиг.15Б приведена таблица, иллюстрирующая пример информации, содержащей элементы данных, проиллюстрированные на Фиг.15А;

на Фиг.16 изображена диаграмма, на которой проиллюстрирован пример структуры запоминающего устройства для информации об идентификаторе объекта согласно варианту осуществления настоящего изобретения;

на Фиг.17 изображена диаграмма, на которой проиллюстрирован пример способа

извлечения информации общей службы обмена сообщениями (GMS) в устройстве обработки информации об идентификаторе объекта согласно варианту осуществления настоящего изобретения; и

на Фиг.18 изображена диаграмма, на которой проиллюстрирован пример способа извлечения сообщаемой информации при получении агентом сообщения Trap (сообщения реакции на необычную ситуацию) из прикладной программы согласно варианту осуществления настоящего изобретения.

Понятно, что на всех чертежах одинаковые номера ссылок относятся к одинаковым узлам, компонентам и структурам.

ПОДРОБНОЕ ОПИСАНИЕ ВАРИАНТОВ ОСУЩЕСТВЛЕНИЯ ИЗОБРЕТЕНИЯ, КОТОРЫЕ ПРИВЕДЕНЫ В КАЧЕСТВЕ ПРИМЕРОВ

Ниже приведено подробное описание вариантов осуществления настоящего изобретения, которые приведены в качестве примеров, со ссылкой на сопроводительные чертежи. В приведенном ниже описании подробные описания содержащихся в них функций и конфигураций, которые являются хорошо известными для специалистов в данной области техники, опущены для доходчивости и краткости изложения.

В вариантах осуществления настоящего изобретения агент простого протокола управления сетью (SNMP) содержит четыре составные части, как показано на Фиг.4.

На Фиг.4 изображена блок-схема, на которой проиллюстрирован пример агента 102 протокола SNMP согласно варианту осуществления настоящего изобретения. Агент 102 протокола SNMP выполняет функцию обеспечения поддержки протокола SNMP в элементе сети.

Агент 102 протокола SNMP содержит агент 400, предназначенный для обработки сообщения простого протокола управления сетью (SNMP) и для осуществления взаимодействия с прикладными программами, и устройство 402 обработки информации об идентификаторе объекта (OIDInfo), предназначенное для обработки информации OIDInfo, соответствующей библиотеке с информацией, для создания общей службы обмена сообщениями (GMS), необходимой для обмена информацией между прикладными программами и агентом 400.

Кроме того, агент 102 протокола SNMP содержит средство 404 извлечения, предназначенное для передачи информации, предназначенной для создания сообщения общей службы обмена сообщениями (GMS) и базы управляющей информации (MIB), в запоминающее устройство для файла OIDInfo, показанное на Фиг.5, и запоминающее устройство 406 для заголовочного файла, предназначенное для запоминания заголовочного файла интерфейса протокола SNMP, используемого для обмена информацией с прикладной программой, информации, управление которой осуществляет прикладная программа 504, показанная на Фиг.5, и информации общей службы обмена сообщениями (GMS).

Устройство 402 обработки информации OIDInfo обеспечивает несколько функций.

Во-первых, при вводе информации OIDInfo об элементах административного управления из агента 400 устройство 402 обработки информации OIDInfo извлекает информацию общей службы обмена сообщениями (GMS), соответствующую информации для обмена ею между агентом 400 и прикладной программой 504, из запоминающего устройства 500 для файла OIDInfo, в котором запомнен файл данных с информацией OIDInfo, и предоставляет информацию общей службы обмена сообщениями (GMS) агенту 400. Здесь, информация, предназначенная для передачи, содержит информацию о номере порта, о формате сообщения, о типе связи, о

структуре полезной информации и т.д., предназначенную для обмена ею между агентом 400 и прикладной программой 504. В варианте осуществления настоящего изобретения, который приведен в качестве примера, предполагают, что интерфейсом между агентом 400 и прикладной программой 504 является общая служба обмена сообщениями (GMS), однако настоящее изобретение не ограничено этим вариантом. В альтернативном варианте могут быть использованы иные интерфейсы.

На Фиг.5 изображена блок-схема, на которой проиллюстрирован агент протокола SNMP, показанный на Фиг.4, и общая последовательность выполнения операций, включающая в себя время компиляции и время выполнения программы, согласно варианту осуществления настоящего изобретения.

Во-первых, агент 400 используют для обработки простого протокола управления сетью (SNMP) с администратором 100, и он производит передачу/прием сообщения общей службы обмена сообщениями (GMS) для обмена информацией с прикладной программой 504 для обеспечения реальной поддержки и осуществления административного управления элементами административного управления. Когда администратор сети передает сообщение с запросом согласно протоколу SNMP, содержащее протокольный блок данных (PDU), на основании различных заранее установленных форматов для выдачи запроса на получение сведений об элементе административного управления сетевого устройства, в котором функционирует агент 102, с использованием администратора 100, то в устройство 402 обработки информации OIDInfo передают идентификатор объекта (OID), поставленный в соответствие элементу административного управления, который включен в состав сообщения с запросом согласно протоколу SNMP. Затем агент 400 получает из устройства 402 обработки информации OIDInfo информацию общей службы обмена сообщениями (GMS), соответствующую информации о номере порта, о формате и о структуре полезной информации для обмена информацией с прикладной программой 504. Агент 400 производит обмен информацией с прикладной программой 504 с использованием информации общей службы обмена сообщениями (GMS) и получает из прикладной программы 504 результат обработки элемента административного управления. После получения результата обработки элемента административного управления из прикладной программы 504 агент 400 посылает данные о результате обработки элемента административного управления администратору 100 посредством ответного сообщения протокола SNMP.

Во-вторых, агент 400 дополнительно используют при получении информации OIDInfo из устройства 402 обработки информации OIDInfo и при передаче протокольного блока данных (PDU) реакции на особую ситуацию (trap) администратору 100 при получении уведомительного (trap) сообщения из прикладной программы 504.

Для доступа к элементу административного управления устройство 402 обработки информации OIDInfo считывает из запоминающего устройства 500 для файла OIDInfo файл с информацией OIDInfo, в котором хранится информация, необходимая для передачи сообщения в прикладную программу 504 и для приема сообщения из нее. Устройство 402 обработки информации OIDInfo извлекает файл информации OIDInfo и передает информацию общей службы обмена сообщениями (GMS) агенту 400. В запоминающем устройстве 500 для файла OIDInfo хранятся прикладные программные интерфейсы (API) библиотеки информации OIDInfo и файлы данных с информацией OIDInfo. Прикладные программные интерфейсы (API) библиотеки информации OIDInfo и файлы данных с информацией OIDInfo предоставляют из

средства 404 извлечения.

Средство 404 извлечения используют при генерации файла базы управляющей информации (MIB) из заголовочного файла интерфейса протокола SNMP, определенного как элемент административного управления в прикладной программе 504, при генерации файла информации OIDInfo, затребованного устройством 402 обработки информации OIDInfo, и при запоминании файла базы управляющей информации (MIB) и файла информации OIDInfo в запоминающем устройстве 500 для файла OIDInfo.

Заголовочный файл 506 интерфейса протокола SNMP представляет собой файл, в котором определен идентификатор (ID) сообщения, номер порта, структура сообщения и т.д., поэтому агент 400 может получить информацию, административное управление которой осуществляют в прикладной программе 504. Заголовочный файл 506 интерфейса протокола SNMP определен в прикладной программе 504 и запомнен в запоминающем устройстве 406 для заголовочного файла.

Ниже приведено описание примера функционирования агента 102 протокола SNMP во время компиляции со ссылкой на Фиг.5, Фиг.6 и Фиг.7.

Теперь будет приведено описание базовой структуры для создания файла данных с информацией OIDInfo, содержащего информацию о сообщении общей службы обмена сообщениями (GMS), подлежащем передаче в прикладную программу 504, и процесса генерации данных об информации OIDInfo на основании этой структуры в средстве 404 извлечения.

С использованием описанного выше способа средство 404 извлечения используют при генерации данных, полученных из запоминающего устройства 406 для заголовочного файла. Агент 400 использует библиотеку информации OIDInfo, будучи способным извлекать необходимую информацию посредством файла информации OIDInfo, и использует информацию о сообщении общей службы обмена сообщениями (GMS), подлежащую передаче в прикладную программу 504.

Теперь будет приведено более подробное описание операции компиляции агента 102 протокола SNMP со ссылкой на Фиг.6. На Фиг.6 проиллюстрирован пример способа генерации файла данных в средстве 404 извлечения, имеющемся в агенте 102 протокола SNMP, во время компиляции согласно варианту осуществления настоящего изобретения.

Сначала разработчик агента 102 протокола SNMP сохраняет заголовочный файл 506 интерфейса протокола SNMP, определяющий информацию, подлежащую административному управлению посредством прикладной программы 504, в запоминающем устройстве 406 для заголовочного файла. Затем средство 404 извлечения считывает заголовочный файл 506 интерфейса протокола SNMP из запоминающего устройства 406 для заголовочного файла с информацией, подлежащей административному управлению, создает файл базы управляющей информации (MIB), в котором определен элемент административного управления, и сохраняет созданный файл базы управляющей информации (MIB) в запоминающем устройстве 502 для файла базы управляющей информации (MIB). Кроме того, средство 404 извлечения считывает заголовочный файл 506 интерфейса протокола SNMP из запоминающего устройства 406 для заголовочного файла, создает файл данных с информацией OIDInfo для генерации сообщения общей службы обмена сообщениями (GMS) и сохраняет файл данных с информацией OIDInfo в запоминающем устройстве 500 для файла OIDInfo.

На Фиг.7 изображена схема последовательности операций, на которой

проиллюстрирован пример функционирования агента 102 протокола SNMP во время компиляции, согласно варианту осуществления настоящего изобретения.

Когда при операции 700 разработчиком выдан запрос на изменение элемента административного управления агента 102 протокола SNMP, в последовательности операций переходят к выполнению операции 702, при которой прикладная программа 504 создает заголовочный файл 506 интерфейса протокола SNMP для измененного элемента административного управления на основании информации, введенной разработчиком, и сохраняет созданный заголовочный файл 506 интерфейса протокола SNMP в запоминающем устройстве 406 для заголовочного файла.

При операции 704 средство 404 извлечения считывает заголовочный файл интерфейса протокола SNMP и создает базу управляющей информации (MIB) и файл данных с информацией OIDInfo. При операции 706 из созданных при выполнении операции 704 базы управляющей информации (MIB) и файла данных с информацией OIDInfo базу управляющей информации (MIB) запоминают в запоминающем устройстве 502 для файла базы управляющей информации (MIB), а созданный файл данных с информацией OIDInfo запоминают в запоминающем устройстве 500 для файла OIDInfo.

На Фиг.8 проиллюстрирован пример способа, выполняемого в агенте 102 протокола SNMP во время выполнения программы с использованием файла данных с информацией OIDInfo, генерация которого осуществлена во время компиляции, показанной на Фиг.6.

При обмене информацией между агентом 400 и прикладной программой 504, показанной на Фиг.8, используют интерфейс общей службы обмена сообщениями (GMS) (или объектно-ориентированной службы обмена сообщениями). Как показано на Фиг.10, полезная информация 1009 сообщения общей службы обмена сообщениями (GMS) содержит сведения о типе сообщения для поддержки обмена информацией по протоколу SNMP между агентом 400 и прикладной программой 504 и заголовок EM_Interface_header 1002, указывающий количество повторяющихся структур в полезной информации сообщения 1009. Более подробное его описание приведено ниже.

Сначала, когда администратор 100 передает агенту 400 сообщение с запросом согласно протоколу SNMP, агент 400 передает информацию OIDInfo в сообщении с запросом согласно протоколу SNMP в устройство 402 обработки информации OIDInfo. Одна из причин того, почему агент 400 передает информацию OIDInfo в устройство 402 обработки информации OIDInfo, состоит в том, что для агента 400 требуется обмен информацией общей службы обмена сообщениями (GMS) с прикладной программой 504.

Устройство 402 обработки информации OIDInfo считывает информацию общей службы обмена сообщениями (GMS) на основании принятого сообщения с идентификатором объекта (OID) и передает считанную информацию общей службы обмена сообщениями (GMS) агенту 400. Затем агент 400 производит обмен информацией с прикладной программой 504 на основании информации общей службы обмена сообщениями (GMS). На основании результирующего ответа прикладной программы 504 агент 400 передает администратору 100 ответ на запрос простого протокола управления сетью (SNMP).

Теперь приведено описание примера функционирования компонентов агента 102 протокола SNMP во время выполнения программы со ссылкой на Фиг.9.

На Фиг.9 изображена схема последовательности операций, на которой

проиллюстрировано функционирование агента 102 протокола SNMP во время выполнения программы, согласно варианту осуществления настоящего изобретения.

При операции 900 агент 400 определяет, был ли принят запрос протокола SNMP от администратора 100. Если при операции 900 запрос протокола SNMP был принят, то агент 400 переходит к выполнению операции 902, при которой он передает идентификатор объекта (OID), содержащийся в сообщении с запросом согласно протоколу SNMP, принятом от администратора 100, в устройство 402 обработки информации OIDInfo.

При операции 904 устройство 402 обработки информации OIDInfo считывает на основании принятой информации OIDInfo соответствующую информацию общей службы обмена сообщениями (GMS) из запоминающего устройства 500 для файла OIDInfo и предоставляет эту информацию общей службы обмена сообщениями (GMS) агенту 400. Информация общей службы обмена сообщениями (GMS) содержит здесь информацию прикладной программы 504 о номере порта, о типе обмена информацией, о структуре данных и т.д., предназначенную для обмена ею между агентом 400 и прикладной программой 504.

При операции 906 агент 400, получающий информацию общей службы обмена сообщениями (GMS), передает в прикладную программу 504 запрос на получение данных об элементе административного управления, запрошенных администратором 100 на основании информации общей службы обмена сообщениями (GMS). При операции 908 прикладная программа 504 посылает данные об элементе административного управления посредством ответного сообщения общей службы обмена сообщениями (GMS).

При операции 910 агент 400 передает администратору 100 ответное сообщение протокола SNMP на основании данных об элементе административного управления, соответствующих ответу из прикладной программы.

На Фиг.10 проиллюстрирован пример конфигурации протокольного блока данных (PDU) общей службы обмена сообщениями (GMS) для обмена информацией между агентом 400 и прикладной программой 504. Для поддержки обмена информацией по простому протоколу управления сетью (SNMP) между всеми программами с использованием протокола SNMP требуется заголовок 1000 общей службы обмена сообщениями (GMS). Полезная информация 1009 общей службы обмена сообщениями (GMS) содержит заголовок EM_Interface_header 1002 для специального заголовочного файла для обеспечения поддержки обмена информацией по протоколу SNMP между агентом 400 и прикладной программой 504.

Заголовок EM_Interface_header 1002 содержит четыре поля (более подробное описание которых приведено ниже со ссылкой на Фиг.11). Заголовок EM_Interface_header 1002 используют для обеспечения поддержки обмена информацией по протоколу SNMP между агентом 400 и прикладной программой 504. Поле 1004 "структура полезной информации" содержит значение поля полезной информации, необходимой во время обмена информацией по протоколу SNMP.

На Фиг.11-Фиг.14 проиллюстрирован пример конфигурации сообщения с протокольным блоком данных PDU, и ниже приведено более подробное описание полей конфигурации в соответствии с типом сообщения, передаваемого между агентом 400 и прикладной программой 504.

Сначала приведено краткое описание конфигурации и функции соответствующих полей.

Набор из заголовка общей службы обмена сообщениями (GMS) (GMS Hdr) 1000,

заголовка EM_Interface_header 1002 и поля 1004 "структура" именуют термином "одна строка 1014".

Теперь будет приведено более подробное описание заголовка EM_Interface_header 1002, состоящего, как описано выше, из четырех описанных выше полей. Первое поле 1006 msgType ("тип сообщения") указывает тип сообщения, передаваемого между агентом 400 и прикладной программой 504, а поле 1008 rowCount ("количество строк") указывает количество повторяемых структур 1004 в полезной информации во время операций Get, Set и Get-Next для таблицы, поддерживающей multiRowTable (таблицу с множеством строк). Во время операции Get-Bulk для всех таблиц поле 1008 rowCount ("количество строк") используют для цели, идентичной той, которую выполняет значение Max-Repeat (максимальное количество повторов) при операции Get-Bulk протокола SNMP. Ответ 1010 используют для сообщения о возникновении ошибки. Поле 1012 structId ("идентификатор структуры") указывает идентификатор сообщения, дополнительно используемый в прикладной программе 504, и его используют в качестве идентификатора связи (relation ID) для данных с программируемой загрузкой (PLD).

Кроме того, прикладная программа 504 копирует идентификатор transactionId 1016 транзакции и идентификатор bsmId 1018 из заголовка 1000 сообщения общей службы обмена сообщениями (GMS), принятого от агента 400, создает ответное сообщение и передает созданное ответное сообщение агенту 400 таким образом, что агент 400 функционирует в обычном режиме согласно ответу 1010, принятому из прикладной программы 504.

На Фиг.11 проиллюстрирован пример процедуры передачи сообщения запроса Get/GetNext/GetFirst из агента 400 в прикладную программу 504 и операции, связанной с ответом.

Агент 400 выделяет объем памяти, соответствующий одной строке, вставляет значение индекса, принятое от администратора 100, в полезную информацию общей службы обмена сообщениями (GMS) 1009, и передает полезную информацию 1009 общей службы обмена сообщениями (GMS) в прикладную программу 504.

Прикладная программа 504 копирует в запоминающее устройство строку, соответствующую индексу, принятому из агента 400 в соответствующей таблице, и передает ответ агенту 400. В это время значение MsgId (идентификатора сообщения) устанавливают равным значению ResponseId (идентификатора ответа), соответствующего заранее заданной структуре. Если произошел сбой, то вставляют и передают значение ошибки, связанное с ответом 1010 в виде заголовка EM_Interface_header 1002. В это время производят повторную передачу полезной информации 1009 общей службы обмена сообщениями (GMS) без ее видоизменения.

В случае наличия запроса GetNextRequest производят извлечение и передачу следующей строки, идущей после соответствующей строки.

На Фиг.12 проиллюстрирован пример процедуры передачи сообщения PreSet/SetRequest из агента 400 в прикладную программу 504 и операции, связанной с ответом.

Сначала производят передачу сообщения PreSet/SetRequest из агента 400 в прикладную программу 504. Функционирование запроса PreSetRequest является, по существу, таким же самым, как и в случае запроса GetRequest. После получения запроса PreSetRequest прикладная программа 504 фиксирует соответствующую строку. После получения ответа на запрос PreSetRequest из прикладной программы 504

агент 400 изменяет значение атрибута для выполнения запроса SetRequest и передает запрос SetRequest в прикладную программу 504.

После получения обычного ответа из прикладной программы 504 агент 400 передает администратору 100 ответ на сообщение Set ("установить"). Если из прикладной программы 504 принято сообщение Fail ("неудача"), то агент 400 передает сообщение Fail ("неудача") администратору 100. Когда время превышено, то агент 400 аннулирует пакет Set ("установить").

В случае запроса SetRequest выделяют объем памяти, соответствующий строке, для запрошенной полезной информации 1009 общей службы обмена сообщениями (GMS), а затем передают полезную информацию 1009 общей службы обмена сообщениями (GMS). Когда прикладная программа передает ответ, то агент 400 передает только заголовок EM_Interface_header 1002.

На Фиг.13 проиллюстрирован пример процедуры передачи сообщения GetBulkRequest для считывания нескольких значений объекта из агента 400 в прикладную программу 504 и операции, связанной с ответом.

В случае запроса GetBulkRequest из соответствующей строки осуществляют генерацию нескольких областей памяти в соответствии со значением rowCount ("количество строк") (Max-Repeat ("максимальное количество повторов")) 1008 из заголовка EM_Interface_header 1002. Например, если значение rowCount ("количество строк") равно "0", то передают всю таблицу.

Если агентом 400 уже был передан в прикладную программу 504 запрос, в котором значение rowCount ("количество строк") равно, например, "3", то для запрошенной полезной информации 1009 общей службы обмена сообщениями (GMS) выделяют только лишь такой объем памяти, который соответствует одной строке, для передачи индекса, указывающего начало. Затем прикладная программа 504 стирает описанный выше объем памяти, выделяет несколько областей памяти, соответствующих значению поля 1008 rowCount ("количество строк"), копирует информацию и передает скопированную информацию агенту 400. Агент 400 стирает память.

На Фиг.14 проиллюстрирован пример процедуры передачи сообщения "уведомление" (Trap) из агента 400 в прикладную программу 504 и операции, связанной с ответом.

Для обработки уведомления в предпочтительном варианте должна быть определена заранее заданная структура, и предоставляют идентификатор одиночного сообщения. Прикладная программа 504 устанавливает значение в поле msgType ("тип сообщения") заголовка EM_Interface_header 1002, равное EM_NOTIFICATION, заполняет заранее заданное значение идентификатора сообщения msgId 1400 в заголовке 1000 общей службы обмена сообщениями (GMS), заполняет информацию о структуре в полезной информации 1009 и передает сообщение агенту 400.

Теперь приведено описание примера файла данных с информацией OIDInfo, хранящегося в запоминающем устройстве 500 для файла OIDInfo.

На Фиг.15А и Фиг.15Б схематично проиллюстрированы таблицы, в которых файл данных с информацией OIDInfo поставлен в соответствие области памяти (на чертеже не показана) в запоминающем устройстве 500 для файла OIDInfo.

На Фиг.15А схематично проиллюстрированы примеры элементов данных, содержащих файл с информацией об идентификаторе объекта. Описание соответствующих элементов данных приведено со ссылкой на Фиг.15Б.

На Фиг.15Б приведена таблица, иллюстрирующая пример информации, содержащей элементы данных, проиллюстрированные на Фиг.15А.

Во-первых, заголовок OIDInfoHdr 1500 информации об идентификаторе объекта (OID) указывает структуру данных полезной информации, хранящей основные сведения об информации OIDInfo. Эта информация о структуре содержит сведения о версии и дату генерации информации OIDInfo, идентификатор

объекта (OID), заданный по умолчанию, значение смещения для указания самого верхнего узла в информации OIDTreeInfo 1502 о древовидной структуре идентификатора объекта, которое служит в качестве информации о структуре каждого узла дерева идентификаторов объекта (OID), и значения смещения для указания заголовка NotiInfoHdr 1508 сообщаемой информации, используемого для извлечения сообщаемой информации.

Подробное описание полей, содержащих заголовки OIDInfoHdr 1500 информации идентификатора объекта, приведено в таблице 1, которая служит в качестве примера.

Таблица 1	
Наименование поля	Описание
Версия	Это поле указывает версию файла информации OIDInfo об идентификаторе объекта.
Дата	Это поле указывает дату генерации файла информации OIDInfo.
DefaultOID (идентификатор объекта, заданный по умолчанию)	Это значение содержится во всех идентификаторах объекта (OIDs) и исключается во время поиска в информации OIDInfo об идентификаторе объекта.
OIDTreeInfoOffset	Это поле указывает значение смещения для указания самого высокого OIDTreeInfo в полях OIDTreeInfo.
NotiInfoHdrOffset	Это поле указывает значение смещения для указания заголовка NotiInfoHdr самого высокого уровня в полях NotiInfoHdr для извлечения сообщаемой информации с использованием идентификатора сообщения.

Информация OIDTreeInfo 1502 содержит информацию об узле дерева структуры и используется для поиска по дереву. Информацию OIDTreeInfo 1502 используют для выражения группы узлов и для выражения одного узла информации OIDTreeInfo 1502, соответствующего скалярному объекту и табличному объекту. Эта структура содержит данные ObjectID, указывающие идентификатор объекта (OID) базы управляющей информации (MIB), тип узла (NodeType) и четыре смещения.

Данные ObjectID указывают идентификатор объекта (OID) текущего узла базы управляющей информации (MIB), а не общий идентификатор объекта (OID). Данные NodeType ("тип узла") используют для определения того, является ли текущий узел групповым объектом, скалярным объектом или табличным объектом. Смещение, содержащееся в информации OIDTreeInfo содержит смещение, указывающее более высокий узел OIDTreeInfo и смещение, указывающее поле GMSInfo с информацией об общей службе обмена сообщениями (GMS) и о структуре.

Описание полей, содержащихся в информации OIDTreeInfo 1502, приведено в таблице 2, которая служит в качестве примера.

Таблица 2	
Наименование поля	Описание
objectId (идентификатор объекта)	Это поле указывает идентификатор объекта (OID) соответствующего узла дерева. Значение идентификатора объекта (OID) является целым числом, указывающим идентификатор объекта (OID) соответствующего узла, а не общий идентификатор объекта (OID).
nodeType (тип узла)	Это поле используют для определения того, является ли тип узла дерева скаляром, таблицей или группой.
upOIDTreeInfoOffset	Это значение представляет собой значение смещения, указывающее более высокий узел OIDTreeInfo в дереве.
nextOIDTreeInfoOffset	Это значение представляет собой значение смещения для указания следующего узла OIDTreeInfo на том же самом уровне в дереве идентификатора объекта (OID).

gmsInfoOffset	Это значение представляет собой значение смещения относительно информации общей службы обмена сообщениями (GMS) в том случае, когда типом узла является скаляр или таблица. Если типом узла (nodeType) является скаляр или таблица и смещение gmsInfoOffset равно 0, или если типом узла (nodeType) является группа и смещение gmsInfoOffset не равно 0, то это означает, что генерация файла информации OIDInfo средством извлечения была произведена неправильно.
---------------	---

5

Информация общей службы обмена сообщениями (GMS) (GMSInfo) 1504 содержит информацию заголовка общей службы обмена сообщениями (GMS) и полезную информацию для передачи сообщения в прикладную программу 504. Эта структура разделена на поля, значения которых изменяются вместе с полем structType ("тип структуры") и общими полями. Идентификатор объекта (OID) поставлен в соответствие группе, включающей в себя скаляр или таблицу базы управляющей информации (MIB), генерация которой осуществлена с использованием этой структуры, и содержит общий идентификатор объекта (OID), заданный по умолчанию (или идентификатор объекта (OID) в масштабе предприятия). Данные structname ("наименование структуры") указывают наименование, созданное при определении структуры. Поле structType ("тип структуры") используют для указания 6 различных типов структуры, а поле payloadType ("тип полезной информации") представляет собой поле для указания того, возможны ли операции Set, Get и Get-Next для множества строк с использованием этой структуры.

Идентификатор requestMsgId сообщения с запросом, идентификатор responseMsgId ответного сообщения и portNumber ("номер порта") указывают информацию заголовка общей службы обмена сообщениями (GMS). Поле numberOfIndex указывает количество индексов структуры. Поле numberOfField указывает общее количество полей структуры, а не количество атрибутов таблицы базы управляющей информации (MIB). Поле payloadSize указывает общий размер структуры для вычисления объема полезной информации общей службы обмена сообщениями (GMS). Идентификатор pldRelationId указывает идентификатор связи для данных с программируемой загрузкой (PLD), а поле masterTableOffset обозначает смещение для указания главной таблицы, когда structType ("тип структуры") является подтаблица. Поле nextGMSInfoOffset представляет собой зарезервированное поле для использования в будущем, и назначение этого поля в настоящее время не определено. Наконец, поле firstGMSAttInfoOffset обозначает смещение для указания первого поля GMSAttInfo, указывающего информацию о полях структуры.

Поля информации GMSInfo 1504 используют для различных целей согласно данным structType ("типу структуры"). Идентификатор requestMsgId сообщения с запросом, идентификатор responseMsgId ответного сообщения и portNumber ("номер порта") обычно используют в качестве информации заголовка общей службы обмена сообщениями (GMS). Назначение данных nextGMSInfoOffset и firstGMSAttInfoOffset является, по существу, идентичным между всеми полями structType ("тип структуры"). В таблице 3 в качестве примера приведено описание данных structType ("тип структуры") и назначения полей, изменяющихся при изменении данных structType ("тип структуры").

Таблица 3	
Тип	Описание
SType_Scalar (скаляр)	Это поле указывает структуру, содержащую скаляры.
SType_StaticTable (статическая таблица)	Эта таблица указывает статическую таблицу для административного управления данными в прикладной программе.
SType_AgentDyTable (динамическая таблица агента)	Эта таблица указывает динамическую таблицу для административного управления данными в агенте.

50

SType_AppDyTable (динамическая таблица прикладной программы)	Эта таблица указывает динамическую таблицу для административного управления данными в прикладной программе.
SType_PLD (данные с программируемой загрузкой)	Эта таблица указывает таблицу данных с программируемой загрузкой (PLD). Операции Get, Get-Next или Get-Bulk позволяют агенту получать данные через интерфейс услуг маршрутизации (RSI), и в прикладную программу передают только команду Set.
SType_SubTable (подтаблица, массив)	Это поле указывает информацию GMSInfo общей службы обмена сообщениями (GMS), расширенную за счет использования массива.

Теперь приведено описание примера назначения полей GMSInfo 1504 в соответствии с данными structType ("тип структуры").

Сначала, в таблице 4, которая приведена в качестве примера, показано назначение каждого поля GMSInfo 1504 в том случае, когда structType ("типом структуры") является скаляр.

Таблица 4		
Наименование поля	Описание	Примечание
OID	Это поле указывает строку идентификатора объекта (OID) до группы, содержащей скалярные узлы.	
structname	Это поле указывает строку наименования структуры, определенную в заголовочном файле интерфейса протокола SNMP.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
structType	Это поле указывает тип структуры информации GMSInfo общей службы обмена сообщениями (GMS). Типом структуры является скаляр. (structType=SType_Scalar)	
payloadType	Это поле указывает, возможны ли одновременные операции Set, Get и Get-Next для множества строк с использованием структуры. Оно может быть использовано только тогда, когда прикладная программа обеспечивает функцию отмены действия. Имеется два типа: тип "одиночная строка" и тип "множество строк".	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
requestMsgId	Это поле указывает идентификатор сообщения запроса, вставленный в заголовок общей службы обмена сообщениями (GMS) при передаче агентом запроса в прикладную программу.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
responseMsgId	Это поле указывает идентификатор ответного сообщения, вставленный в заголовок общей службы обмена сообщениями (GMS) при передаче ответа прикладной программой агенту.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
portNumber	Это поле указывает номер порта, в котором прикладная программа получает сообщение от агента.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
numberOfIndex	Это поле указывает количество индексов структуры. В данном случае количество индексов структуры равно 0. Когда собраны скаляры, то индекс не существует.	
numberOfField	Это поле указывает количество полей структуры, определенной в заголовочном файле интерфейса протокола SNMP. Поле numberOfField содержит сведения о количестве полей индексов. Структура определена следующим образом: struct { intA; //index intB; //index intC; intD[3]; intE[4]; }AA;	
	В описанном выше случае в поле numberOfField указано значение 5.	
payloadSize	Это поле указывает общий объем структуры.	
pIdRelationId	Это поле не используют, и оно имеет нулевое значение.	

masterTableOffset	Это поле не используют, и оно имеет нулевое значение.	
nextGMSInfoOffset	Это поле не определено и является зарезервированным для его использования в будущем.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
firstGMSAttInfoOffset	Это поле указывает значение смещения для указания первого поля GMSAttInfo, указывающего информацию о полях структуры.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.

В следующей таблице 5, которая приведена в качестве примера, показано назначение каждого поля GMSInfo 1504 в том случае, когда типом структуры (structType) является статическая таблица.

Таблица 5		
Наименование поля	Описание	Примечание
OID	Это поле указывает строку идентификатора объекта (OID) до элемента таблицы базы управляющей информации (MIB).	
structname	Это поле указывает строку наименования структуры, определенную в заголовочном файле интерфейса протокола SNMP.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
structType	Это поле указывает тип структуры информации GMSInfo общей службы обмена сообщениями (GMS). Типом структуры является статическая таблица. (structType=SType_StaticTable)	
payloadType	Это поле указывает, возможны ли одновременные операции Set, Get и Get-Next для множества строк с использованием структуры. Оно может быть использовано только тогда, когда прикладная программа обеспечивает функцию отмены действия. Имеется два типа: тип "одиночная строка" и тип "множество строк".	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
requestMsgId	Это поле указывает идентификатор сообщения запроса, вставленный в заголовок общей службы обмена сообщениями (GMS) при передаче агентом запроса в прикладную программу.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
responseMsgId	Это поле указывает идентификатор ответного сообщения, вставленный в заголовок общей службы обмена сообщениями (GMS) при передаче ответа прикладной программой агенту.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
portNumber	Это поле указывает номер порта, в котором прикладная программа получает сообщение от агента.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
numberOfIndex	Это поле указывает количество индексов структуры.	
numberOfField	Это поле указывает количество полей структуры, определенной в заголовочном файле интерфейса протокола SNMP. Если структура содержит массив, то в этом поле содержатся сведения о количестве полей, указывающих массив. (База управляющей информации (MIB) не имеет поля, указывающего массив.)	
payloadSize	Это поле указывает общий объем структуры.	
pldRelationId	Это поле не используют, и оно имеет нулевое значение.	
masterTableOffset	Это поле не используют, и оно имеет нулевое значение.	
nextGMSInfoOffset	Это поле не определено и является зарезервированным для его использования в будущем.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
firstGMSAttInfoOffset	Это поле указывает значение смещения для указания первого поля GMSAttInfo, указывающего информацию о полях структуры.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.

В следующей таблице 6, которая приведена в качестве примера, показано назначение каждого поля GMSInfo 1504 в том случае, когда типом структуры (structType) является динамическая таблица агента.

Таблица 6

Наименование поля	Описание	Примечание
OID	Это поле указывает строку идентификатора объекта (OID) до элемента таблицы базы управляющей информации (MIB).	
structname	Это поле указывает строку наименования структуры, определенную в заголовочном файле интерфейса протокола SNMP.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
structType	Это поле указывает тип структуры информации GMSInfo общей службы обмена сообщениями (GMS). Типом структуры является динамическая таблица агента. (structType=SType_AgentDyTable)	
payloadType	Это поле указывает, возможны ли одновременные операции Set, Get и Get-Next для множества строк с использованием структуры. Оно может быть использовано только тогда, когда прикладная программа обеспечивает функцию отмены действия. Имеется два типа: тип "одиночная строка" и тип "множество строк".	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
requestMsgId	Это поле указывает идентификатор сообщения запроса, вставленный в заголовок общей службы обмена сообщениями (GMS) при передаче агентом запроса в прикладную программу.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
responseMsgId	Это поле указывает идентификатор ответного сообщения, вставленный в заголовок общей службы обмена сообщениями (GMS) при передаче ответа прикладной программой агенту.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
portNumber	Это поле указывает номер порта, в котором прикладная программа получает сообщение от агента.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
numberOfIndex	Это поле указывает количество индексов структуры.	
numberOfField	Это поле указывает количество полей структуры, определенной в заголовочном файле интерфейса протокола SNMP. (В базе управляющей информации (MIB) существуют данные RowStatus ("состояние строки"), но поле, поставленное в соответствие данным RowStatus, не определено при определении структуры. Данные GMSAttInfo информации OIDInfo не существуют.Административное управление данными RowStatus осуществляет агент.)	
payloadSize	Это поле указывает общий объем структуры.	
pldRelationId	Это поле не используют, и оно имеет нулевое значение.	
masterTableOffset	Это поле не используют, и оно имеет нулевое значение.	
nextGMSInfoOffset	Это поле не определено и является зарезервированным для его использования в будущем.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
firstGMSAttInfoOffset	Это поле указывает значение смещения для указания первого поля GMSAttInfo, указывающего информацию о полях структуры.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.

В следующей таблице 7, которая приведена в качестве примера, показано назначение каждого поля GMSInfo 1504 в том случае, когда типом структуры (structType) является динамическая таблица прикладной программы.

Таблица 7

Наименование поля	Описание	Примечание
OID	Это поле указывает строку идентификатора объекта (OID) до элемента таблицы базы управляющей информации (MIB).	

	structname	Это поле указывает строку наименования структуры, определенную в заголовочном файле интерфейса протокола SNMP.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
5	structType	Это поле указывает тип структуры информации GMSInfo общей службы обмена сообщениями (GMS). Типом структуры является динамическая таблица прикладной программы.	
10	payloadType	Это поле указывает, возможны ли одновременные операции Set, Get и Get-Next для множества строк с использованием структуры. Оно может быть использовано только тогда, когда прикладная программа обеспечивает функцию отмены действия. Имеется два типа: тип "одиночная строка" и тип "множество строк".	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
15	requestMsgId	Это поле указывает идентификатор сообщения запроса, вставленный в заголовок общей службы обмена сообщениями (GMS) при передаче агентом запроса в прикладную программу.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
	responseMsgId	Это поле указывает идентификатор ответного сообщения, вставленный в заголовок общей службы обмена сообщениями (GMS) при передаче ответа прикладной программой агенту.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
20	portNumber	Это поле указывает номер порта, в котором прикладная программа получает сообщение от агента.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
	numberOfIndex	Это поле указывает количество индексов структуры.	
25	numberOfField	Это поле указывает количество полей структуры, определенной в заголовочном файле интерфейса протокола SNMP. Поскольку в структуре содержится поле, поставленное в соответствие данным RowStatus ("состояние строки"), то в случае динамической таблицы прикладной программы в него включены сведения о количестве полей. Данные GMSAttInfo информации OIDInfo существуют.Административное управление данными RowStatus осуществляет прикладная программа.	
	payloadSize	Это поле указывает общий объем структуры.	
30	pldRelationId	Это поле не используют, и оно имеет нулевое значение.	
	masterTableOffset	Это поле не используют, и оно имеет нулевое значение.	
	nextGMSInfoOffset	Это поле не определено и является зарезервированным для его использования в будущем.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
35	firstGMSAttInfoOffset	Это поле указывает значение смещения для указания первого поля GMSAttInfo, указывающего информацию о полях структуры.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.

В следующей таблице 8, которая приведена в качестве примера, показано назначение каждого поля GMSInfo 1504 в том случае, когда типом структуры (structType) являются данные с программируемой загрузкой (PLD).

	Таблица 8		
	Наименование поля	Описание	Примечание
45	OID	Это поле указывает строку идентификатора объекта (OID) до элемента таблицы базы управляющей информации (MIB).	
	structname	Это поле указывает строку наименования структуры, определенную в заголовочном файле интерфейса протокола SNMP.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
50	structType	Это поле указывает тип структуры информации GMSInfo общей службы обмена сообщениями (GMS). Типом структуры являются данные с программируемой загрузкой (PLD).	

5	payloadType	Это поле указывает, возможны ли одновременные операции Set, Get и Get-Next для множества строк с использованием структуры. Оно может быть использовано только тогда, когда прикладная программа обеспечивает функцию отмены действия. Имеется два типа: тип "одиночная строка" и тип "множество строк".	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
	requestMsgId	Это поле указывает идентификатор сообщения запроса, вставленный в заголовок общей службы обмена сообщениями (GMS) при передаче агентом запроса в прикладную программу.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
10	responseMsgId	Это поле указывает идентификатор ответного сообщения, вставленный в заголовок общей службы обмена сообщениями (GMS) при передаче ответа прикладной программой агенту.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
	portNumber	Это поле указывает номер порта, в котором прикладная программа получает сообщение от агента.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
15	numberOfIndex	Это поле указывает количество индексов структуры.	
	numberOfField	Это поле указывает количество полей структуры. Когда структура содержит массив, то это поле содержит сведения о количестве полей массива. (Поле массива не находится в базе управляющей информации (MIB).) Например, когда количество полей равно 3, то значение numberOfField равно 3.	
20	payloadSize	Это поле указывает общий объем структуры.	
	pldRelationId	Это поле указывает идентификатор связи (relation ID) для данных с программируемой загрузкой (PLD).	
	masterTableOffset	Это поле не используют, и оно имеет нулевое значение.	
25	nextGMSInfoOffset	Это поле не определено и является зарезервированным для его использования в будущем.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
	firstGMSAttInfoOffset	Это поле указывает значение смещения для указания первого поля GMSAttInfo, указывающего информацию о полях структуры.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.

30 В следующей таблице 9, которая приведена в качестве примера, показано назначение каждого поля GMSInfo 1504 в том случае, когда типом структуры (structType) является подтаблица.

			Таблица 9
35	Наименование поля	Описание	Примечание
	OID	Это поле указывает строку идентификатора объекта (OID) до элемента таблицы базы управляющей информации (MIB).	
40	structname	Это поле указывает строку наименования структуры, определенную в заголовочном файле интерфейса протокола SNMP.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
	structType	Это поле указывает тип структуры информации GMSInfo общей службы обмена сообщениями (GMS). Типами структуры являются скаляр, статическая таблица, динамическая таблица агента, динамическая таблица прикладной программы, данные с программируемой загрузкой (PLD) и подтаблица.	
45			
	payloadType	Это поле указывает, возможны ли одновременные операции Set, Get и Get-Next для множества строк с использованием структуры. Оно может быть использовано только тогда, когда прикладная программа обеспечивает функцию отмены действия. Имеется два типа: тип "одиночная строка" и тип "множество строк".	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
50			

	requestMsgId	Это поле указывает идентификатор сообщения запроса, вставленный в заголовок общей службы обмена сообщениями (GMS) при передаче агентом запроса в прикладную программу.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
5	responseMsgId	Это поле указывает идентификатор ответного сообщения, вставленный в заголовок общей службы обмена сообщениями (GMS) при передаче ответа прикладной программой агенту.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
	portNumber	Это поле указывает номер порта, в котором прикладная программа получает сообщение от агента.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
10	numberOfIndex	Это поле указывает количество индексов подтаблицы, а не индексов главной таблицы в базе управляющей информации (MIB). Значение numberOfIndex указывает порядок массива.	
15	numberOfField	Это поле указывает количество полей для указание индекса и значение расширенной таблицы, когда расширение произведено посредством массива. Например, значение numberOfField может быть равно 2.	
	payloadSize	Это поле указывает объем типа информации, определяющего массив.	
20	pldRelationId	Это поле не используют, и оно имеет нулевое значение.	
	masterTableOffset	Это поле указывает исходную главную таблицу, содержащую массив, указывающий подтаблицу.	
	nextGMSInfoOffset	Это поле не определено и является зарезервированным для его использования в будущем.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.
25	firstGMSAttInfoOffset	Это поле указывает значение смещения для указания первого поля GMSAttInfo, указывающего информацию о полях структуры.	Это поле имеет одинаковый смысл вне зависимости от значения поля structType.

Информация о полях, вставленная в полезную информацию, может быть обнаружена при последовательном извлечении информации GMSAttInfo, указанной посредством поля firstGMSAttInfoOffset.

Поскольку описание поля и структуры для информации GMSInfo 1504 уже было приведено выше, то теперь будет описана структура информации GMSAttInfo 1506. GMSAttInfo 1506 указывает информацию о поле структуры, которая подлежит вставке в полезную информацию общей службы обмена сообщениями (GMS). Элементы GMSAttInfo 1506, указывающие соответствующие поля структуры, связаны друг с другом согласно смещениям. Значение смещения для последнего элемента GMSAttInfo равно "0".

Когда структура содержит массив, то по порядку следования полей GMSAttInfo 1506 первым оказывается поле GMSAttInfo, указывающее индекс, а последним оказывается поле GMSAttInfo, указывающее массив.

Индексы расположены в порядке следования идентификаторов объекта (OID), и значения также расположены в порядке следования идентификаторов объекта (OID). Значения полей GMSAttInfo отличаются в соответствии с structType ("типом структуры") информации GMSInfo и с fieldType ("типом поля") информации GMSAttInfo.

Сначала будет приведено описание информации fieldType ("тип поля"), определяющей тип поля GMSAttInfo. Примеры назначения полей, которые являются различными в соответствии с fieldType ("типом поля"), показаны в таблице 10, которая приведена в качестве примера.

Таблица 10		
Тип	Описание	Примечание

	FType_ScalarValue (скалярное значение)	Это поле GMSAttInfo указывает скалярное значение.	
	FType_TableValue (табличное значение)	Это поле GMSAttInfo указывает значение атрибута таблицы.	
	FType_TableAndIndex (таблица и индекс)	Это поле GMSAttInfo указывает значение индекса таблицы.	
5	FType_TableAndArray (таблица и массив)	Это поле GMSAttInfo указывает поле, определяемое массивом, когда определена структура.	
10	FType_ArrayAndIndex (массив и индекс)	Это поле GMSAttInfo указывает поле индекса, когда structType ("тип структуры") более высокого поля GMSInfo является подтаблицей. Если в том случае, когда structType ("типом структуры") является не подтаблица, появляется тип поля "массив и индекс", то это означает, что файл информации OIDInfo был неправильно сконфигурирован средством извлечения.	

Поля GMSAttInfo 1506 в том случае, когда structType ("типом структуры") является не подтаблица, а fieldType ("типом поля") является скалярное значение, табличное значение или "таблица и индекс", показаны в таблице 11, которая приведена в качестве примера.

Таблица 11		
Наименование поля	Описание	Примечание
objectId	Это поле указывает идентификатор объекта (OID) скаляра или атрибута таблицы.	
attName	Это поле указывает строку наименования поля структуры, определенной в заголовочном файле интерфейса протокола SNMP.	
asnType	Это поле указывает тип синтаксиса, выраженный в базе управляющей информации (MIB).	
accessType	Это поле указывает тип доступа, выраженный в базе управляющей информации (MIB).	
fieldType	Это поле указывает тип поля (например, скалярное значение, табличное значение или "таблица и индекс").	
attType	Это поле указывает тип данных на языке Си, определенный в информации о структуре.	
attSize	Это поле указывает размер поля.	
startOffset	Это поле указывает начальное смещение для поля в полезной информации общей службы обмена сообщениями (GMS) и указывает начальное смещение для информации о начале структуры.	
subTable	Это поле не используют, и оно имеет нулевое значение.	
maxDimensionValue	Это поле не используют, и оно имеет нулевое значение.	
nextAttInfoOffset	Это поле указывает значение смещения для указания следующего поля GMSAttInfo.	Это поле имеет одинаковый смысл вне зависимости от значения fieldType.

Поля GMSAttInfo 1506 в том случае, когда structType ("типом структуры") является не подтаблица, а fieldType ("типом поля") является "таблица и массив", показаны в таблице 12, которая приведена в качестве примера.

Таблица 12		
Наименование поля	Описание	Примечание
objectId	Это поле не используют, поскольку для таблицы, указывающей массив, было произведено расширение. Значение objectId установлено равным 0.	
attName	Это поле указывает строку наименования поля структуры, определенной в заголовочном файле интерфейса протокола SNMP.	
asnType	Значение этого поля является фиксированным и равным ASN_Null	
accessType	Значение этого поля является фиксированным и равным AType_NotAccessible.	
fieldType	Это поле указывает тип поля ("таблица и массив").	

5	attType	Это поле указывает тип данных на языке Си, определенный в информации о структуре.	
	attSize	Это поле указывает размер поля и указывает общий размер поля, указывающего массив. Например, attSize=200, если поле определено как "int a[10][5];".	
	startOffset	Это поле указывает начальное смещение для поля в полезной информации общей службы обмена сообщениями (GMS).	
	subTable	Это поле указывает значение смещения для указания информации GMSInfo, расширенной посредством массива.	
	maxDimensionValue	Это поле не используют, и оно имеет нулевое значение.	
10	nextAttInfoOffset	Это поле указывает значение смещения для указания следующего поля GMSAttInfo.	Это поле имеет одинаковый смысл вне зависимости от значения fieldType.

Поля GMSAttInfo 1506 в том случае, когда structType ("типом структуры") является не подтаблица, а fieldType ("типом поля") является "массив и индекс", показаны в таблице 13, которая приведена в качестве примера.

Таблица 13		
Наименование поля	Описание	Примечание
20 objectId	Это поле указывает идентификатор объекта (OID) в базе управляющей информации (MIB).	
	attName	Здесь указано наименование поля, созданное при генерации базы управляющей информации (MIB).
	asnType	Это поле всегда имеет значение ASN_Unsigned32. (Значение начинается с 0.)
25 accessType	Значение этого поля является фиксированным и равным ACType_NotAccessible.	
	fieldType	Это поле указывает тип поля ("массив и индекс").
	attType	Это поле указывает тип данных на языке Си, определенный в информации о структуре. В этом случае тип attType всегда равен AttType_Int.
30 attSize	Это значение всегда равно 4 (поскольку значение int size равно 4).	
	startOffset	Это поле не используют, и оно имеет нулевое значение.
	subTable	Это поле не используют, и оно имеет нулевое значение.
	maxDimensionValue	Это поле указывает максимальное значение индекса и определяет размер массива.
35 nextAttInfoOffset	Это поле указывает значение смещения для указания следующего поля GMSAttInfo.	Это поле имеет одинаковый смысл вне зависимости от значения fieldType.

Поля GMSAttInfo 1506 в том случае, когда structType ("типом структуры") является не подтаблица, а fieldType ("типом поля") является табличное значение, показаны в таблице 14, которая приведена в качестве примера.

Таблица 14		
Наименование поля	Описание	Примечание
45 objectId	Это поле указывает идентификатор объекта (OID) в базе управляющей информации (MIB).	
	attName	Это поле указывает наименование атрибута, созданное при генерации базы управляющей информации (MIB).
	asnType	Это поле указывает тип синтаксиса, выраженный в базе управляющей информации (MIB).
50 accessType	Это поле указывает тип доступа, выраженный в базе управляющей информации (MIB).	
	fieldType	Это поле указывает тип поля ("табличное значение").
	attType	Это поле указывает тип данных на языке Си, определенный в информации о структуре.
	attSize	Это поле указывает размер поля.

startOffset	Это поле указывает начальное смещение для начала массива в полезной информации общей службы обмена сообщениями (GMS). Это поле имеет то же самое значение, как и в том случае, когда structType ("типом структуры") является не подтаблица, а fieldType ("типом поля") является "таблица и массив".	
subTable	Это поле не используют, и оно имеет нулевое значение.	
maxDimensionValue	Это поле не используют, и оно имеет нулевое значение.	
nextAttInfoOffset	Это поле указывает значение смещения для указания следующего поля GMSAttInfo.	

Заголовок NotiInfoHdr 1508 из Фиг.15А и Фиг.15Б представляет собой информацию о структуре, которую используют в устройстве 402 обработки информации OIDInfo для получения сообщаемой информации с использованием идентификатора сообщения. В способе извлечения сообщаемой информации могут быть использованы один или два идентификатора сообщения.

Заголовок NotiInfoHdr 1508 содержит поле, указывающее количество полей NotiNodeInfo 1510, и смещение, указывающее первое поле NotiNodeInfo 1510. Описание полей NotiInfoHdr 1508 приведено в Таблице 15, которая служит в качестве примера.

Таблица 15		
Наименование поля	Описание	Примечание
numberOfNotiNodeInfo	Это поле указывает количество полей NotiNodeInfo, расположенных последовательно в порядке следования идентификаторов сообщения. Поле numberOfNotiNodeInfo используют для извлечения информации NotiNodeInfo при помощи соответствующего идентификатора сообщения.	
firstNotiNodeInfoOffset	Это поле указывает значение смещения первого поля NotiNodeInfo из последовательно расположенных полей NotiNodeInfo.	

Поле NotiNodeInfo 1510 из Фиг.15 содержит поле для запоминания идентификатора сообщения, поле notiNodeType для того, чтобы различать тип NotiNodeInfo, поле subNotiInfoHdrOffset для указания подзаголовка subNotiInfoHdr, и поле notiInfoOffset для указания поля NotiInfo 1512 с сообщаемой информацией.

Поле NotiNodeInfo 1510 представляет собой информацию о структуре, используемую для извлечения соответствующей сообщаемой информации с использованием идентификатора сообщения. Информация NotiNodeInfo 1510 размещена в запоминающем устройстве 500 для файла данных с информацией OIDInfo в порядке следования идентификаторов сообщения. Устройство 402 обработки информации OIDInfo может извлекать соответствующую сообщаемую информацию с использованием поля NotiNodeInfo 1510 и заголовка NotiInfoHdr 1508.

Сначала, когда одному элементу сообщаемой информации поставлен в соответствие один идентификатор сообщения, выполняют алгоритм двоичного поиска с использованием данных numberOfNotiNodeInfo и firstNotiInfoOffset из заголовка NotiInfoHdr. Используя этот алгоритм, производят поиск соответствующего поля NotiNodeInfo 1510. Это возможно потому, что поле NotiNodeInfo 1510 расположено в порядке следования идентификаторов сообщения.

Уведомление простого протокола управления сетью (SNMP) создают с использованием информации NotiInfo 1512, указанной посредством данных notiInfoOffset о смещении в поле NotiNodeInfo 1510.

Затем, когда одному элементу сообщаемой информации поставлены в соответствие два идентификатора сообщения, поиск NotiNodeInfo 1510 производят с использованием

описанного выше способа. В этом случае типом `notiNodeType` информации `NotiNodeInfo` 1510 является узел множества уведомлений. Смещение `subNotiInfoHdrOffset` указывает подзаголовок `sub NotiInfoHdr` 1508.

По существу, идентичный алгоритм повторяют с использованием идентификатора второго сообщения. Используя этот алгоритм, производят поиск `NotiNodeInfo` 1510. Уведомление простого протокола управления сетью (SNMP) создают с использованием информации `NotiInfo`, указанной посредством данных `notiInfoOffset` о смещении в поле `NotiNodeInfo` 1510.

Описание вышеупомянутых полей приведено в Таблице 16, которая служит в качестве примера.

Таблица 16		
Наименование поля	Описание	Примечание
<code>notiMsgId</code>	Это поле указывает идентификатор сообщения для извлечения сообщаемой информации.	
<code>notiNodeType</code>	Это поле указывает тип <code>NotiNodeInfo</code> . Имеется три типа: узел одиночного уведомления, узел множества уведомлений и узел суб-уведомления. Узел одиночного уведомления соответствует тому случаю, когда один идентификатор сообщения поставлен в соответствие одному уведомлению. Узел множества уведомлений соответствует тому случаю, когда сообщаемой информации поставлены в соответствие два идентификатора сообщения. Узел множества уведомлений представляет собой такой тип информации <code>NotiNodeInfo</code> , поиск которой осуществляют с использованием идентификатора первого сообщения. Узел суб-уведомления представляет собой такой тип информации <code>NotiNodeInfo</code> , поиск которой осуществляют с использованием идентификатора второго сообщения.	
<code>subNotiInfoHdrOffset</code>	Это значение представляет собой смещение для указания подзаголовка <code>sub NotiInfoHdr</code> в том случае, когда типом <code>notiNodeType</code> узла является узел множества уведомлений. Когда типом <code>notiNodeType</code> узла является узел одиночного уведомления или узел суб-уведомления, то значение смещения равно 0.	
<code>notiInfoOffset</code>	Это значение представляет собой смещение для указания информации <code>NotiInfo</code> в том случае, когда типом <code>notiNodeType</code> узла является узел одиночного уведомления или узел множества уведомлений. Когда типом <code>notiNodeType</code> узла является узел множества уведомлений, то значение смещения равно 0.	

Теперь будет приведено описание информации `NotiInfo` 1512. Информация `NotiInfo` 1512 содержит данные `notiInfoType` о типе для того, чтобы различать тип уведомления, данные `numberOfNotiField` для указания количества полей в структуре уведомления, которые определены в заголовочном файле интерфейса протокола SNMP, и первое смещение информации о поле сообщения.

Сначала, в таблице 17, которая приведена в качестве примера, показаны данные `notiInfoType` о типе.

Таблица 17		
Тип	Описание	Примечание
<code>Nltype_CommonNoti</code>	Это поле указывает общее уведомление, подлежащее передаче от агента только администратору.	
<code>NIType_InformNoti</code>	После выдачи уведомления агент ожидает приема информирующего сообщения. Если информирующее сообщение от администратора не принято, то агент выполняет заранее заданную процедуру.	

В Таблице 18, которая приведена в качестве примера, показаны поля, содержащиеся в информации `NotiInfo` 1512.

Таблица 18

Наименование поля	Описание	Примечание
NotiOID	Это поле указывает идентификатор объекта (OID) уведомления, определенный в базе управляющей информации (MIB).	
NotiInfoType	Это поле указывает тип уведомления для обработки информационного сообщения.	
NumberOfNotiField	Это поле указывает количество полей структуры уведомления, определенных в заголовочном файле интерфейса протокола SNMP.	
FirstNotiAttInfoOffset	Это поле представляет собой смещение для указания первого поля NotiAttInfo с информацией о полях структуры уведомления, определенной в заголовочном файле интерфейса протокола SNMP.	

Нижe приведено описание данных NotiAttInfo 1514, соответствующих информации о последней структуре. Данные NotiAttInfo 1514 указывают информацию о полях уведомления. Во-первых, имеются данные notiAttOID, представляющие собой строку, указывающую идентификатор объекта (OID) поля для предоставления уведомления. Данные notiAttOID указывают полный идентификатор объекта (OID) в скаляре и указывает значение, из которого был исключен индекс в таблице. Данные notiAttName указывают строку наименования поля, определенного в заголовочном файле интерфейса протокола SNMP. Данные notiASNTType указывают тип синтаксиса атрибута, выраженного в базе управляющей информации (MIB). Данные notiFieldType определяют тип атрибута. Другие поля имеют различное назначение в соответствии с типом notiFieldType поля уведомления. Данные notiAttType указывает тип данных поля на языке Си. Данные notiAttSize указывают размер поля. Данные notiStartOffset представляют собой значение смещения для указания положения поля в информации о структуре, принятой из прикладной программы. Данные subNotiAttOffset представляют собой значение смещения для указания информации NotiAttInfo для выражения массива в том случае, если текущее поле указывает наличие массива. Данные notiMaxDimensionValue указывают максимальное значение индекса, когда типом является "массив и индекс" в данных NotiAttInfo, расширенных за счет использования массива. Оно означает максимальное значение для массива. Наконец, данные nextNotiAttInfoOffset представляют собой значение смещения для указания следующих данных NotiAttInfo 1514. В данных NotiAttInfo 1514 индекс расположен в головной части. Если уведомление содержит множество таблиц, то после значения расположен индекс, и имеются следующий индекс и следующее значение.

Смысл каждого поля является различным в соответствии с типом notiFieldType поля уведомления в информации NotiAttInfo 1514. Это показано в приведенных ниже таблицах. В таблице 19, которая приведена в качестве примера, показан тот случай, когда типом notiFieldType поля уведомления является скалярное значение.

Таблица 19

Наименование поля	Описание	Примечание
notiAttOID	Это поле указывает строку всего идентификатора объекта (OID) скаляра.	
notiAttName	Это поле указывает строку наименования поля структуры.	Это поле имеет одинаковый смысл вне зависимости от значения notiFieldType.
notiASNTType	Это поле указывает тип синтаксиса поля.	
notiFieldType	Это поле указывает тип поля (скалярное значение).	
notiAttType	Это поле указывает тип поля на языке Си.	
notiAttSize	Это поле указывает размер поля.	

	notiStartOffset	Это поле указывает значение смещения для начального поля в структуре уведомления.	Это поле имеет одинаковый смысл вне зависимости от значения notiFieldType.
5	subNotiAttInfoOffset	Это поле представляет собой значение смещения для указания информации NotiAttInfo для каждого массива.	
	maxDimensionValue	Это поле не используют, и оно имеет нулевое значение.	
	nextNotiAttInfoOffset	Это поле указывает значение смещения для указания следующего поля.	Это поле имеет одинаковый смысл вне зависимости от значения notiFieldType.

Следующий случай, в котором типом notiFieldType поля уведомления является "таблица и индекс", показан в таблице 20, которая приведена в качестве примера.

Таблица 20		
Наименование поля	Описание	Примечание
notiAttOID	Это поле указывает строку идентификатора объекта (OID) до атрибута таблицы.	
notiAttName	Это поле указывает строку наименования поля структуры.	Это поле имеет одинаковый смысл вне зависимости от значения notiFieldType.
notiASNTYPE	Это поле указывает тип синтаксиса поля.	
notiFieldType	Это поле указывает тип поля ("таблица и индекс").	
notiAttType	Это поле указывает тип поля на языке Си.	
notiAttSize	Это поле указывает размер поля.	
notiStartOffset	Это поле указывает значение смещения для начального поля в структуре уведомления.	Это поле имеет одинаковый смысл вне зависимости от значения notiFieldType.
subNotiAttInfoOffset	Это поле не используют, и оно имеет нулевое значение.	
maxDimensionValue	Это поле не используют, и оно имеет нулевое значение.	
nextNotiAttInfoOffset	Это поле указывает значение смещения для указания следующего поля.	Это поле имеет одинаковый смысл вне зависимости от значения notiFieldType.

Следующий случай, в котором типом notiFieldType поля уведомления является табличное значение, показан в таблице 21, которая приведена в качестве примера.

Таблица 21		
Наименование поля	Описание	Примечание
notiAttOID	Это поле указывает строку идентификатора объекта (OID) до атрибута таблицы.	
notiAttName	Это поле указывает строку наименования поля структуры.	Это поле имеет одинаковый смысл вне зависимости от значения notiFieldType.
notiASNType	Это поле указывает тип синтаксиса поля.	
notiFieldType	Это поле указывает тип поля (табличное значение).	
notiAttType	Это поле указывает тип поля на языке Си.	
notiAttSize	Это поле указывает размер поля.	
notiStartOffset	Это поле указывает значение смещения для начального поля в структуре уведомления.	Это поле имеет одинаковый смысл вне зависимости от значения notiFieldType.
subNotiAttInfoOffset	Это поле не используют, и оно имеет нулевое значение.	
maxDimensionValue	Это поле не используют, и оно имеет нулевое значение.	
nextNotiAttInfoOffset	Это поле указывает значение смещения для указания следующего поля.	Это поле имеет одинаковый смысл вне зависимости от значения notiFieldType.

Следующий случай, в котором типом notiFieldType поля уведомления является "таблица и массив", показан в таблице 22, которая приведена в качестве примера.

	Таблица 22		
5	Наименование поля	Описание	Примечание
	notiAttOID	Это поле не используют, и его значением является пустая строка.	
	notiAttName	Это поле указывает строку наименования поля структуры.	Это поле имеет одинаковый смысл вне зависимости от значения notiFieldType.
10	notiASNTYPE	Это поле указывает тип синтаксиса поля.	
	notiFieldType	Это поле указывает тип поля ("таблица и массив").	
	notiAttType	Это поле указывает тип поля на языке Си.	
	notiAttSize	Это поле указывает размер поля.	
15	notiStartOffset	Это поле указывает значение смещения для начального поля в структуре уведомления.	Это поле имеет одинаковый смысл вне зависимости от значения notiFieldType.
	subNotiAttInfoOffset	Это поле представляет собой значение смещения для указания информации NotiAttInfo для значения массива.	
	maxDimension Value	Это поле не используют, и оно имеет нулевое значение.	
20	nextNotiAttInfoOffset	Это поле указывает значение смещения для указания следующего поля.	Это поле имеет одинаковый смысл вне зависимости от значения notiFieldType.

Следующий случай, в котором типом notiFieldType поля уведомления в информации NotiAttInfo, указанной посредством смещения subNotiAttInfoOffset в информации NotiAttInfo 1514, является "массив и индекс", показан в Таблице 23, которая приведена в качестве примера.

Таблица 23			
	Наименование поля	Описание	Примечание
30	notiAttOID	Это поле указывает строку идентификатора объекта (OID) до атрибута таблицы в базе управляющей информации (MIB). Эта таблица представляет собой таблицу, расширенную посредством массива.	
	notiAttName	Это поле указывает строку наименования атрибута в базе управляющей информации (MIB).	
35	notiASNType	Это поле всегда имеет значение ASN_Unsigned32. (Значение начинается с 0.)	
	notiFieldType	Это поле указывает тип поля ("массив и индекс").	
	notiAttType	Это поле указывает тип поля на языке Си, и его значение всегда равно AttType_Int.	
40	notiAttSize	Это поле указывает размер поля, и его значение всегда равно 4.	
	notiStartOffset	Это поле не используют, и оно имеет нулевое значение.	
	subNotiAttInfoOffset	Это поле не используют, и оно имеет нулевое значение.	
	maxDimensionValue	Это поле указывает максимальное значение индекса и определяет размер массива.	
45	nextNotiAttInfoOffset	Это поле не используют, и оно имеет нулевое значение.	

Следующий случай, в котором типом notiFieldType поля уведомления в информации NotiAttInfo, указанной посредством смещения subNotiAttInfoOffset в информации NotiAttInfo 1514, является табличное значение, показан в Таблице 24, которая приведена в качестве примера.

Таблица 24		
Наименование поля	Описание	Примечание

	notiAttOID	Это поле указывает строку идентификатора объекта (OID) до атрибута таблицы в базе управляющей информации (MIB). Эта таблица представляет собой таблицу, расширенную посредством массива.	
5	notiAttName	Это поле указывает строку наименования атрибута в базе управляющей информации (MIB).	
	notiASNTType	Это поле указывает тип синтаксиса поля.	
	notiFieldType	Это поле указывает тип поля (табличное значение).	
	notiAttType	Это поле указывает тип поля на языке Си.	
	notiAttSize	Это поле указывает размер поля.	
10	notiStartOffset	Это поле указывает значение смещения для начала поля, указывающего массив в исходной структуре уведомления. Значение смещения notiStartOffset является таким же самым, как и в том случае, когда типом notiFieldType поля уведомления является "таблица и массив".	
	subNotiAttInfoOffset	Это поле не используют, и оно имеет нулевое значение.	
15	maxDimensionValue	Это поле не используют, и оно имеет нулевое значение.	
	nextNotiAttInfoOffset	Это поле не используют, и оно имеет нулевое значение.	

На Фиг.16 изображена диаграмма, на которой проиллюстрирован пример структуры запоминающего устройства 500 для файла OIDInfo. Запоминающее устройство 500 может быть разделено на часть, предназначенную для поиска по дереву с использованием идентификатора объекта (OID), и часть, предназначенную для получения сообщаемой информации с использованием идентификатора сообщения. Одному элементу информации общей службы обмена сообщениями (GMS) поставлена в соответствие одна таблица, выраженная в базе управляющей информации (MIB). В случае скаляра элементы, входящие в состав группы, объединяют, и объединенные элементы ставят в соответствие одному элементу информации общей службы обмена сообщениями (GMS). Также в случае скаляра элементы, входящие в состав группы, объединяют, и объединенные элементы определяют в виде одной структуры, в которой индекс не существует.

Теперь будет приведено описание способа извлечения информации общей службы обмена сообщениями (GMS) согласно варианту осуществления настоящего изобретения для обмена информацией между агентом 400 и прикладной программой 504 со ссылкой на Фиг.17.

На Фиг.17 изображена диаграмма, на которой проиллюстрирован пример способа извлечения информации общей службы обмена сообщениями (GMS) согласно варианту осуществления настоящего изобретения. Когда от администратора 100 принимают запрос в виде сообщений Get, Get-Next, Get-Bulk или Set согласно протоколу SNMP, то устройство 402 обработки информации OIDInfo выполняет описанную ниже процедуру для извлечения соответствующей информации общей службы обмена сообщениями (GMS) с использованием идентификатора объекта (OID), содержащегося в сообщениях Get, Get-Next, Get-Bulk или Set.

Строку идентификатора объекта (OID), введенную из агента 400, отделяют и изменяют на идентификатор объекта (OID) целочисленного типа. Затем производят последовательный поиск информации OIDTreeInfo 1502 с использованием алгоритма поиска по дереву согласно идентификатору объекта (OID).

Затем запоминающее устройство 500 для файла OIDInfo возвращает информацию OIDTreeInfo 1502, поставленную в соответствие идентификатору объекта (OID), и возвращает следующую информацию OIDTreeInfo, идущую после информации OIDTreeInfo 1502, которая поставлена в соответствие идентификатору объекта (OID). Если производят поиск желательной информации OIDTreeInfo, то

устройство 402 обработки информации OIDInfo возвращает информацию GMSInfo 1504, указанную посредством информации OIDTreeInfo. Затем агент 400 создает заголовок общей службы обмена сообщениями (GMS) и полезную информацию с использованием информации GMSInfo 1504. Когда полезная информация 1009 общей службы обмена сообщениями (GMS) создана, то перед полезной информацией вставляют заголовок ОАМ (администрирование и управление операциями), после заголовка ОАМ вставляют заголовок ЕМ для обеспечения поддержки простого протокола управления сетью (SNMP) между агентом 400 и прикладной программой 504, и после заголовка ЕМ вставляют значение структуры. Протокольный блок данных (PDU) общей службы обмена сообщениями (GMS) завершают информацией о полях структуры с использованием данных GMSAttInfo 1506. Затем сообщение передают в прикладную программу 504.

Теперь будет приведено описание способа извлечения сообщаемой информации в агенте 400 согласно варианту осуществления настоящего изобретения со ссылкой на Фиг.18.

На Фиг.18 изображена диаграмма, на которой проиллюстрирован пример способа извлечения сообщаемой информации при получении агентом 400 сообщения Trap (сообщения реакции на необычную ситуацию) из прикладной программы согласно варианту осуществления настоящего изобретения.

Когда агент 400 получает из прикладной программы 504 сообщение Trap (сообщения реакции на необычную ситуацию) с заранее заданным идентификатором сообщения, то устройство 402 обработки информации OIDInfo извлекает сообщаемую информацию протокола SNMP с использованием идентификатора сообщения.

Сначала агент 400 определяет, равно ли количество идентификаторов принятого сообщения одному или двум. Агент 400 проверяет поле subMsgId заголовка 1002 ЕМ. Если значение проверенного поля не равно нулю, то в качестве идентификатора первого сообщения используют идентификатор сообщения заголовка 1000 общей службы обмена сообщениями (GMS), а в качестве идентификатора второго сообщения используют идентификатор subMsgId.

Затем извлекают информацию NotiNodeInfo 1510 с использованием идентификатора первого сообщения. Информацию NotiNodeInfo 1510 извлекают с использованием сведений о количестве полей информации NotiNodeInfo 1510 в заголовке NotiInfoHdr 1508 и алгоритма двоичного поиска. При наличии второго сообщения информацию NotiNodeInfo 1510 извлекают с использованием сведений о количестве полей информации subNotiNodeInfo в заголовке NotiInfoHdr 1508, указанном посредством смещения subNotiInfoHdrOffset в информации NotiNodeInfo 1510, и алгоритма двоичного поиска.

Смещение notiInfoOffset информации NotiNodeInfo 1510 указывает информацию NotiInfo 1512, в которой хранится сообщаемая информация. Уведомительное сообщение протокола SNMP создают с использованием информации NotiInfo 1512 и NotiAttInfo 1514. Созданное сообщение затем посылают администратору 100.

Согласно вариантам осуществления настоящего изобретения, если средство извлечения осуществляет генерацию только нового файла информации OIDInfo, то отсутствует необходимость в том, чтобы агент протокола SNMP добавлял функцию или выполнял новые операции кодирования и повторной компиляции даже в том случае, когда изменен целевой объект административного управления, или когда изменен, добавлен либо удален элемент административного управления. База

управляющей информации (MIB), соответствующая элементу административного управления, и файл информации OIDInfo могут быть автоматически изменены даже в том случае, когда происходит добавление, изменение или удаление структуры, используемой между агентом протокола SNMP и прикладной программой.

Следовательно, может быть легко выполнена разработка агента протокола SNMP, и он может быть использован для других устройств или для других целевых объектов административного управления без видоизменения.

Несмотря на то, что настоящее изобретение было продемонстрировано и описано со ссылкой на определенные варианты его осуществления, которые приведены в качестве примеров, для специалистов в данной области техники понятно, что могут быть выполнены различные изменения, касающиеся его формы и подробностей, не выходя за пределы сущности и объема настоящего изобретения, которые определяются прилагаемой формулой изобретения. Следовательно, объем патентных притязаний настоящего изобретения не следует ограничивать вышеупомянутыми вариантами его осуществления, и он определяется приведенной ниже формулой изобретения и ее эквивалентами.

Формула изобретения

1. Устройство для генерации файла базы управляющей информации (MIB) для административного управления устройством связи с использованием простого протокола управления сетью (SNMP) во время компиляции и файла с информацией об идентификаторе объекта (OIDInfo), предназначенной для обмена ею между агентом протокола SNMP и прикладной программой, содержащее:

запоминающее устройство для заголовочного файла, предназначенное для запоминания заголовочного файла, созданного прикладной программой для административного управления устройством, работающим согласно протоколу SNMP,

средство извлечения, предназначенное для считывания заголовочного файла из запоминающего устройства для заголовочного файла и создания файла базы управляющей информации (MIB) и файла OIDInfo для обмена сообщениями между агентом протокола SNMP и прикладной программой, запоминающее устройство для файла базы управляющей информации (MIB), предназначенное для запоминания файла базы управляющей информации (MIB), созданного средством извлечения, и запоминающее устройство для файла OIDInfo, предназначенное для запоминания файла OIDInfo, созданного средством извлечения.

2. Устройство по п.1, в котором сообщение содержит сообщение общей службы обмена сообщениями (GMS) для обмена информацией между агентом протокола SNMP и прикладной программой.

3. Способ выполнения во время работы операций приема от администратора сообщения с запросом согласно простому протоколу управления сетью протоколу (SNMP), содержащего элемент административного управления, для административного управления устройством связи с использованием протокола SNMP и выдачи ответа о результате обработки элемента административного управления в агенте протокола SNMP, содержащий следующие операции:

в агенте принимают сообщение с запросом согласно протоколу SNMP для запроса на получение данных о результате обработки элемента административного управления, поступившее от администратора, производят передачу информации об идентификаторе объекта (OIDInfo), содержащейся в сообщении с запросом согласно протоколу SNMP, из агента в устройство обработки информации OIDInfo, производят

передачу информации для обмена ею между агентом и прикладной программой, в которой хранятся данные элемента административного управления, запрошенные администратором, из устройства обработки информации OIDInfo агенту на основании информации OIDInfo,

5 производят обмен информацией с прикладной программой в агенте на основании переданной информации, принятой из устройства обработки информации OIDInfo, и принимают результат обработки элемента административного управления, запрошенного администратором, из прикладной программы, и

10 передают принятый результат обработки элемента административного управления из агента администратору.

4. Способ по п.3, в котором операция передачи информации, предназначенной для обмена ею, из устройства обработки информации OIDInfo агенту содержит следующие операции:

15 извлекают информацию общей службы обмена сообщениями (GMS) из файла данных идентификатора объекта (OID), поставленного в соответствие информации OIDInfo, хранящейся в запоминающем устройстве для файла OIDInfo в устройстве обработки информации OIDInfo; и

20 передают информацию общей службы обмена сообщениями (GMS) из устройства обработки информации OIDInfo агенту.

5. Способ по п.4, в котором информация общей службы обмена сообщениями (GMS) содержит информацию, по меньшей мере, об одних из следующих данных: о номере порта, о типе обмена информацией, производимого прикладной программой и о
25 структуре данных полезной информации, предназначенной для обмена между агентом и прикладной программой.

6. Устройство для административного управления устройством связи с использованием SNMP для выполнения во время работы операций приема от администратора сообщения с запросом согласно протоколу SNMP, содержащего
30 элемент административного управления, для административного управления устройством связи с использованием протокола SNMP и выдачи ответа о результате обработки элемента административного управления в агенте SNMP, содержащее: средство обработки информации об идентификаторе объекта (OIDInfo),

35 предназначенное для предоставления информации для обмена ею с прикладной программой при приеме заранее заданной информации OIDInfo, поставленной в соответствие элементу административного управления, и

агента, предназначенного для передачи информации OIDInfo, поставленной в
40 соответствие элементу административного управления, в средство обработки информации OIDInfo при приеме от администратора сообщения с запросом согласно протоколу SNMP, для получения информации, предназначенной для обмена ею с прикладной программой, для приема результата обработки элемента административного управления из прикладной программы и для передачи результата
45 обработки администратору.

7. Устройство по п.6, в котором средство обработки информации OIDInfo содержит: запоминающее устройство для файла OIDInfo, предназначенное для запоминания
50 файла данных с информацией OIDInfo, содержащего информацию для обмена ею между агентом и прикладной программой; и

устройство обработки информации OIDInfo, предназначенное для извлечения информации из файла данных с информацией OIDInfo, хранящегося в запоминающем устройстве для файла OIDInfo, для обмена ею между агентом и прикладной

программой и предназначенное для передачи сообщаемой информации агенту.

8. Устройство по п.6, в котором информация, предназначенная для обмена ею между агентом и прикладной программой, содержит информацию общей службы обмена сообщениями (GMS).

5 9. Устройство по п.8, в котором информация общей службы обмена сообщениями (GMS) содержит информацию, по меньшей мере, об одних из следующих данных: о номере порта, о типе обмена информацией, производимого прикладной программой, и о структуре данных полезной информации, предназначенной для
10 обмена ею между агентом и прикладной программой.

10. Устройство по п.7, в котором запоминающее устройство для файла OIDInfo выполнено таким образом, что способно осуществлять запоминание прикладных программных интерфейсов (API) библиотеки информации OIDInfo и файлов данных с
15 информацией OIDInfo.

20

25

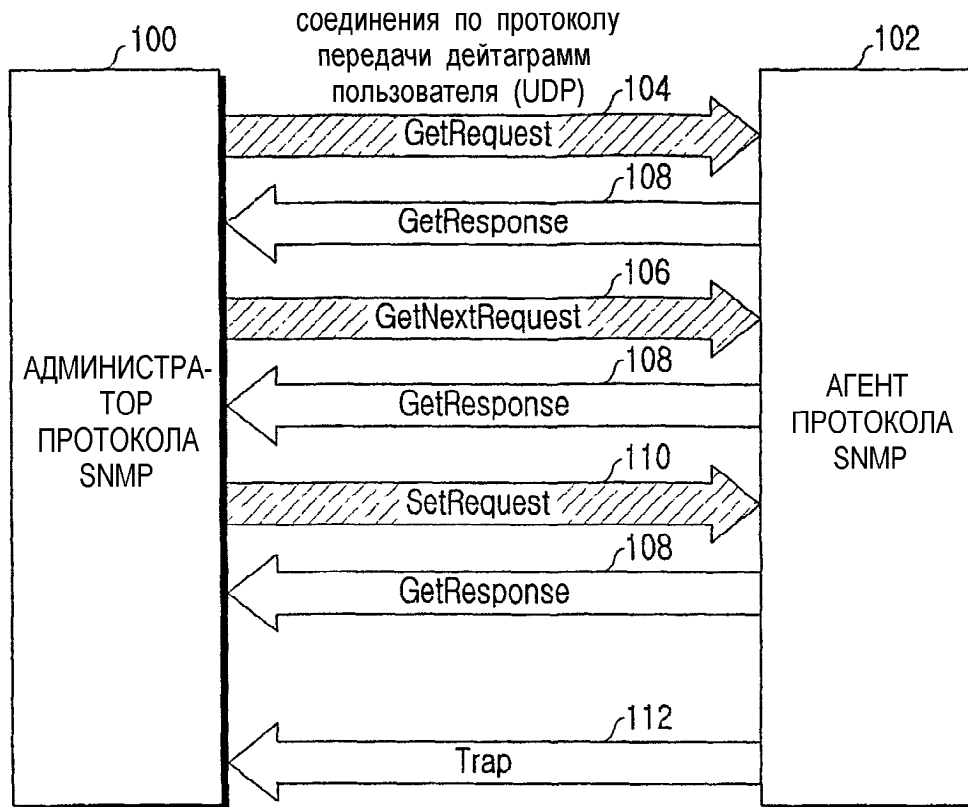
30

35

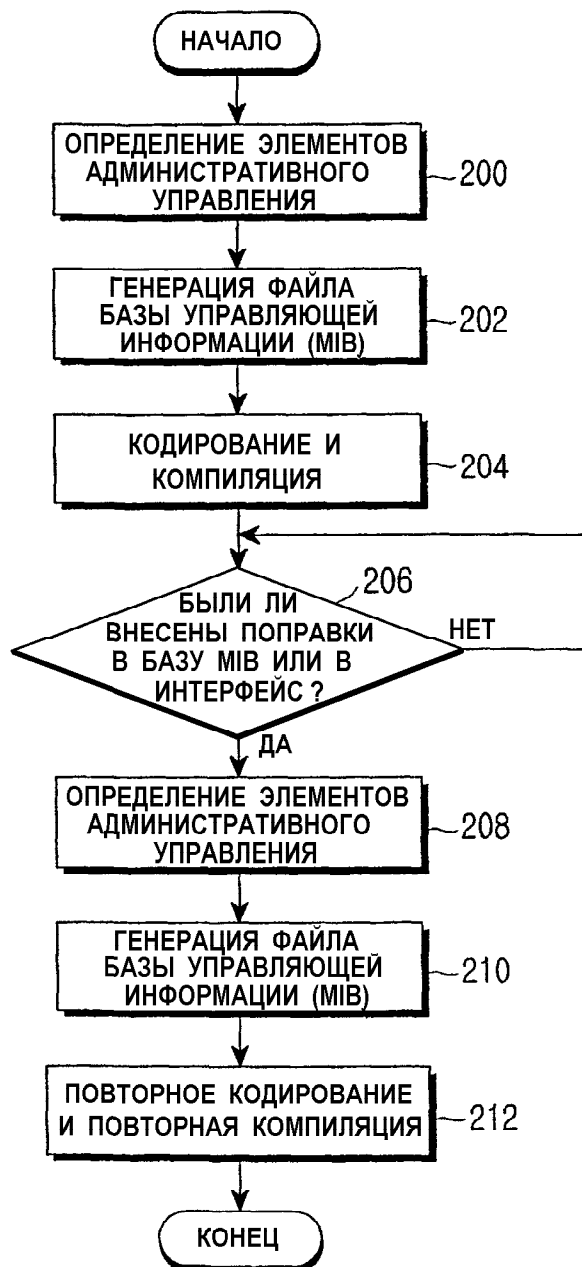
40

45

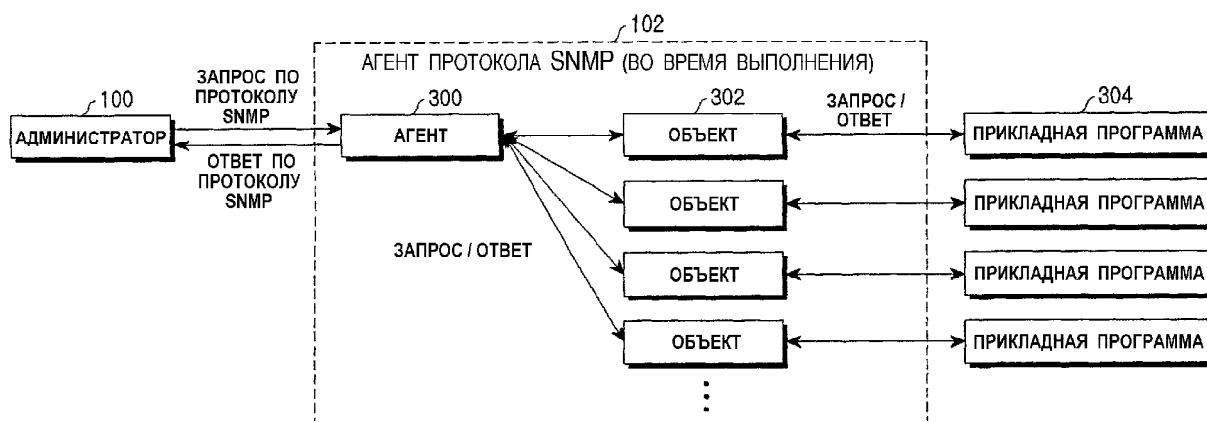
50



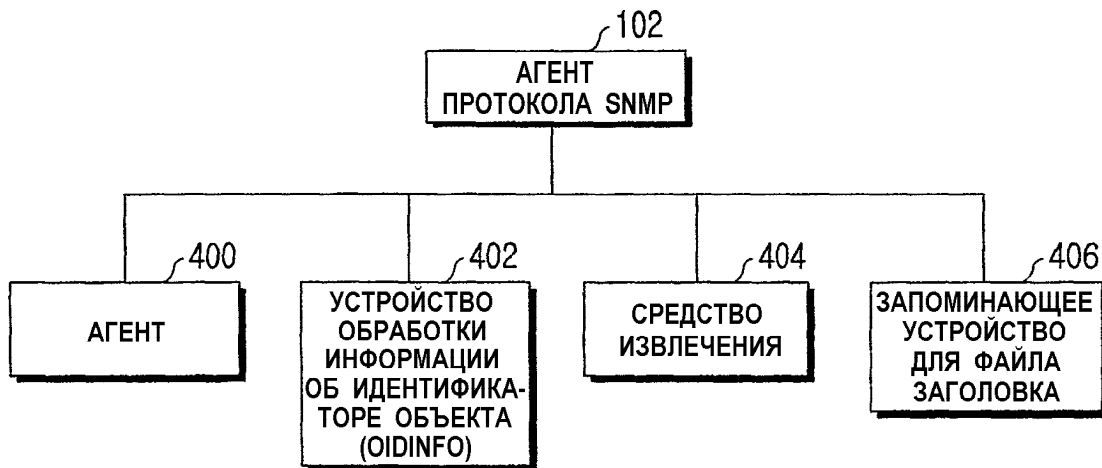
Фиг. 1



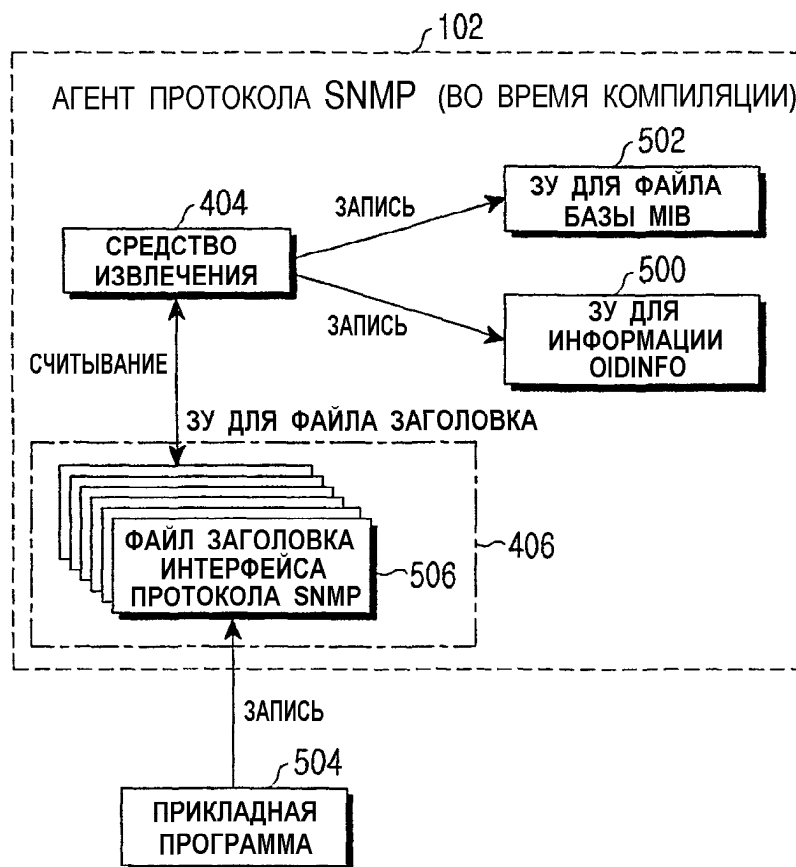
Фиг. 2



Фиг. 3

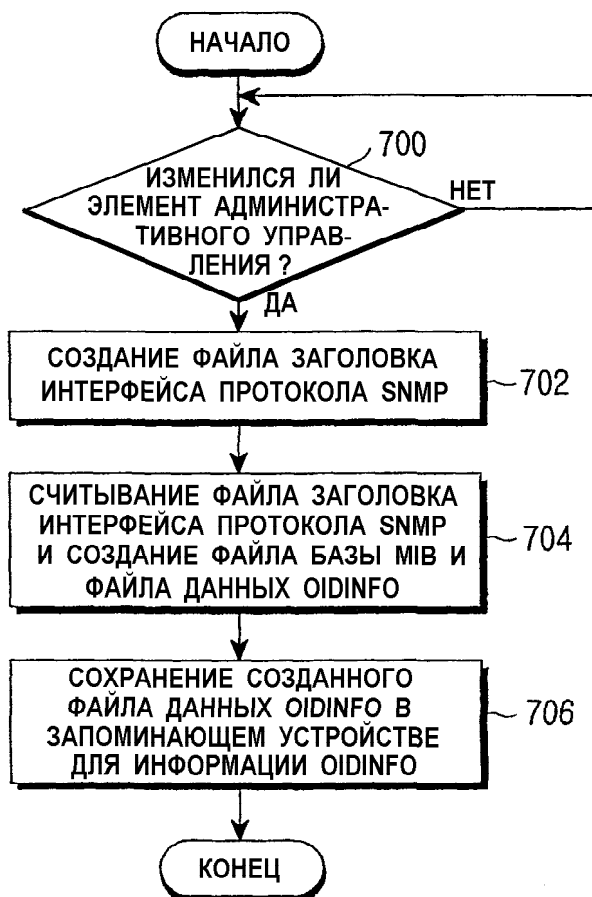


Фиг. 4

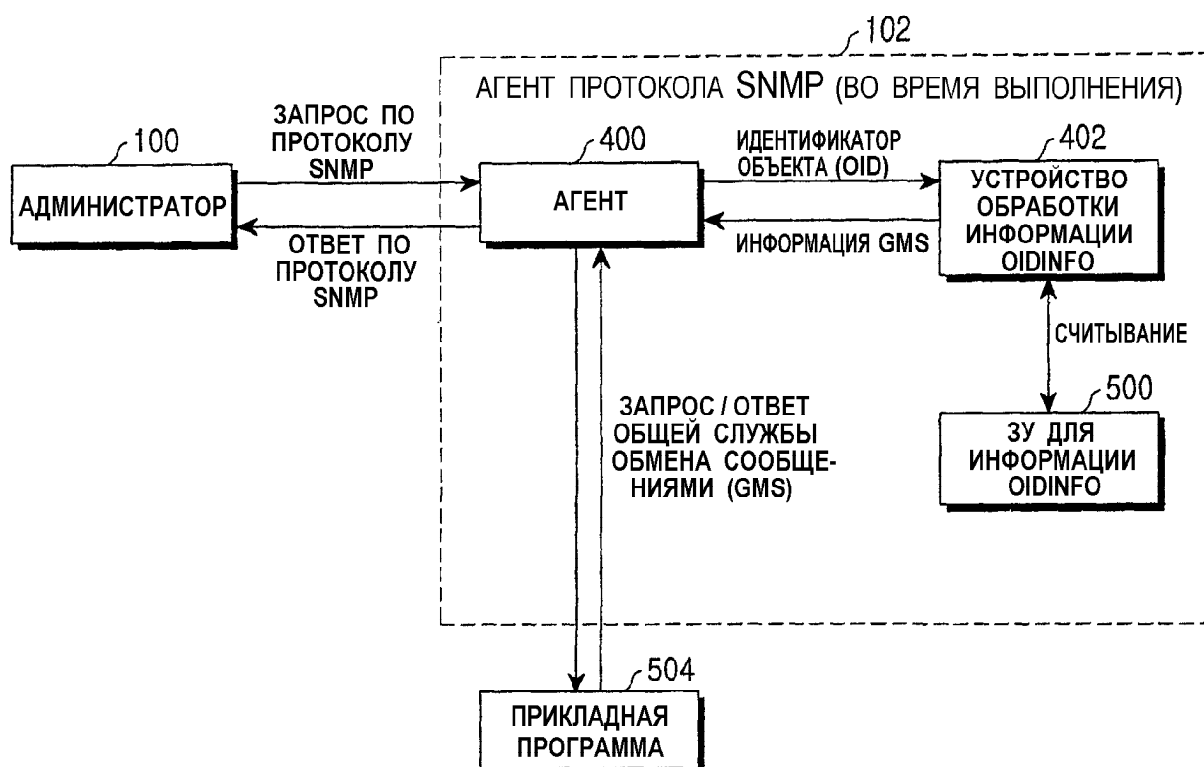


ПРИМЕЧАНИЕ: ЗУ - запоминающее устройство

Фиг. 6

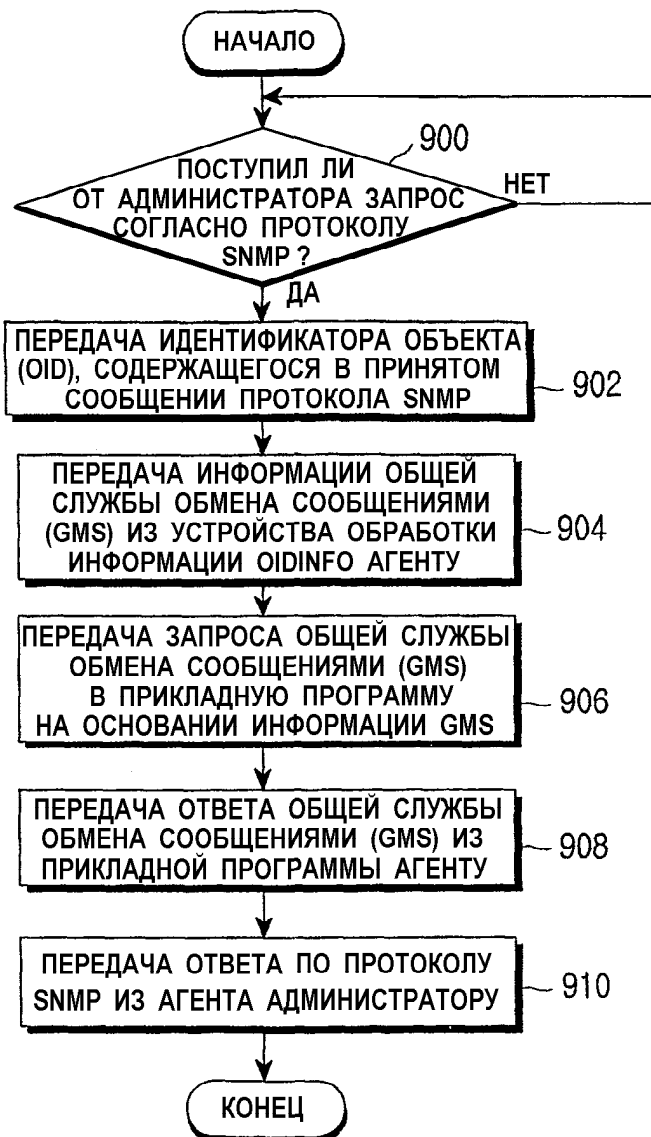


Фиг. 7

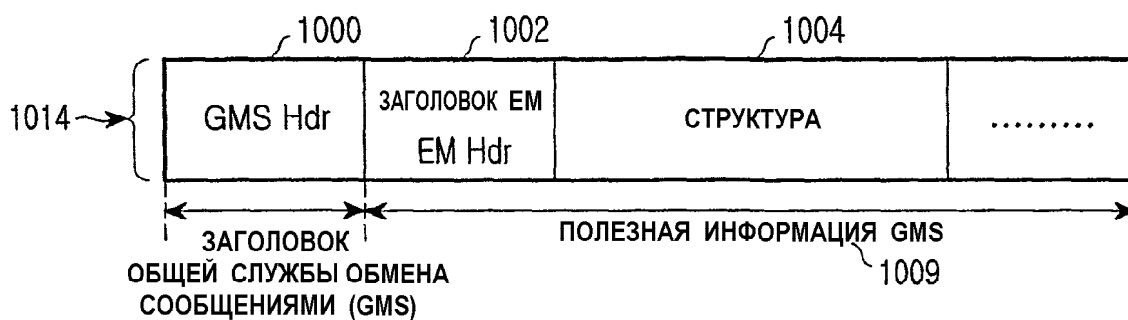


ПРИМЕЧАНИЕ: ЗУ - запоминающее устройство

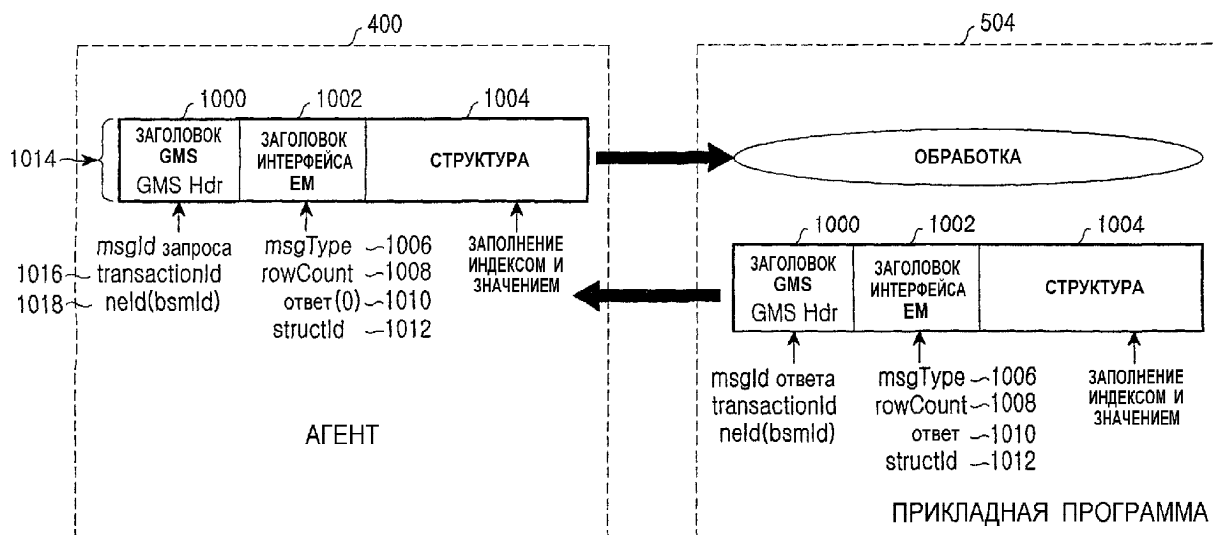
Фиг. 8



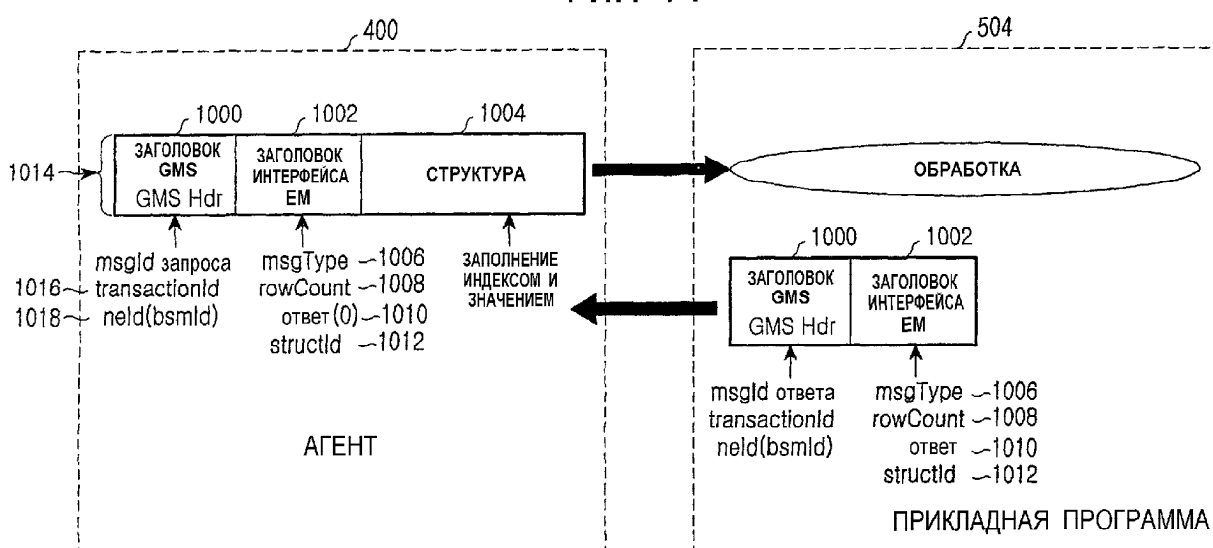
Фиг. 9



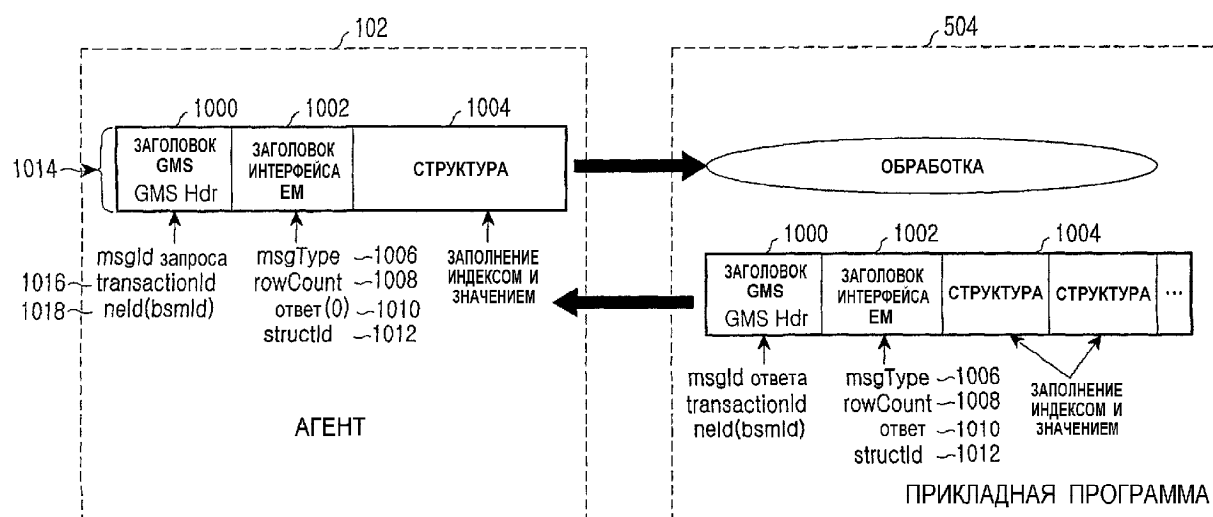
Фиг. 10



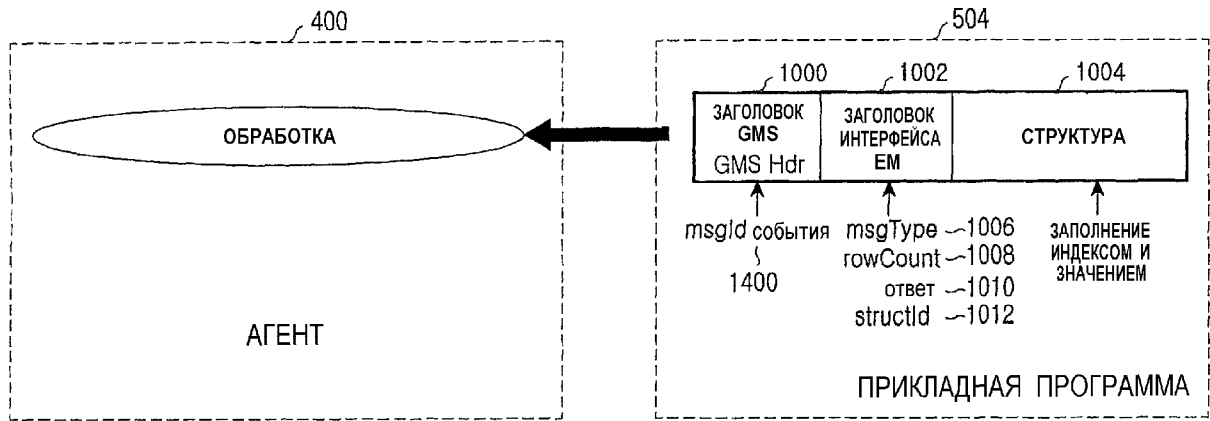
Фиг. 11



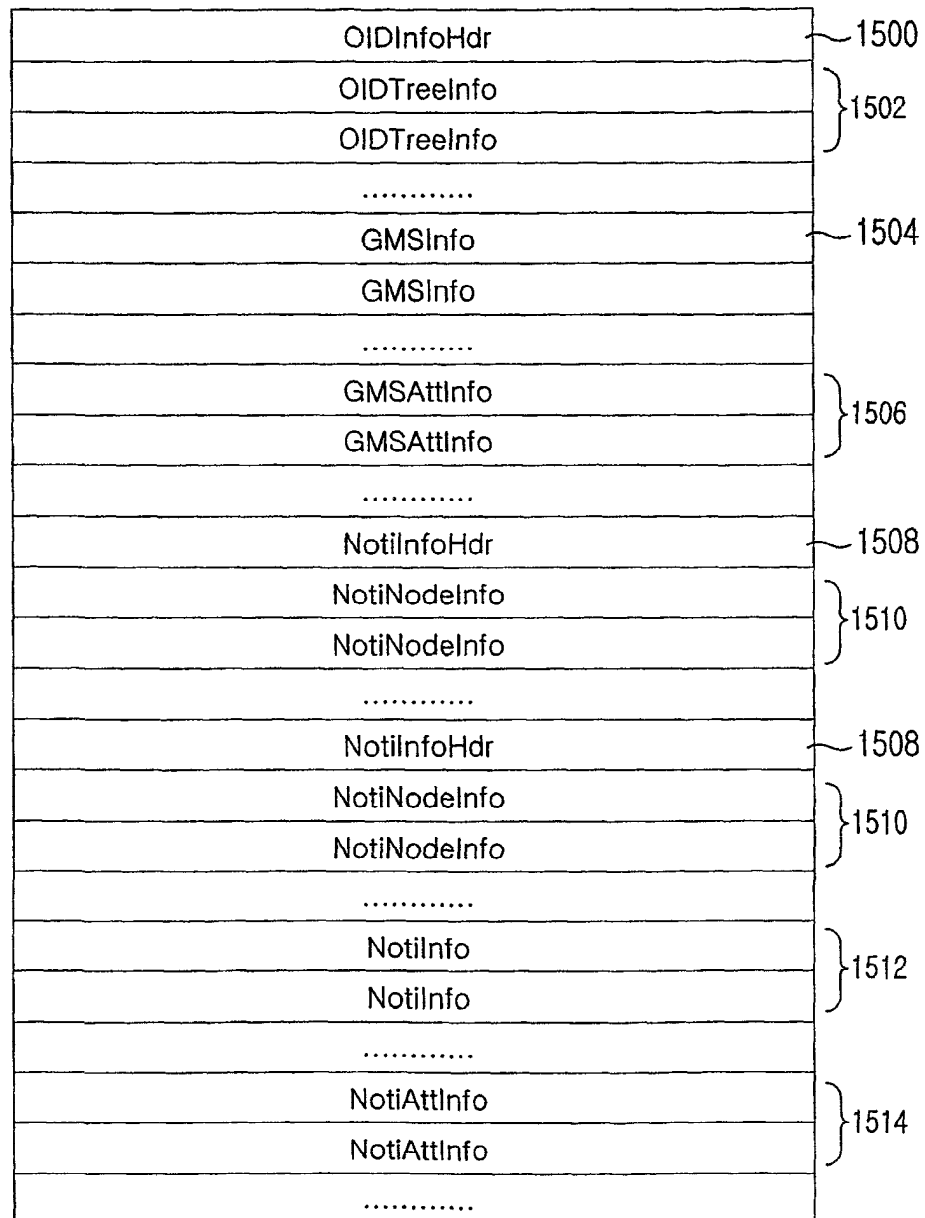
Фиг. 12



Фиг. 13



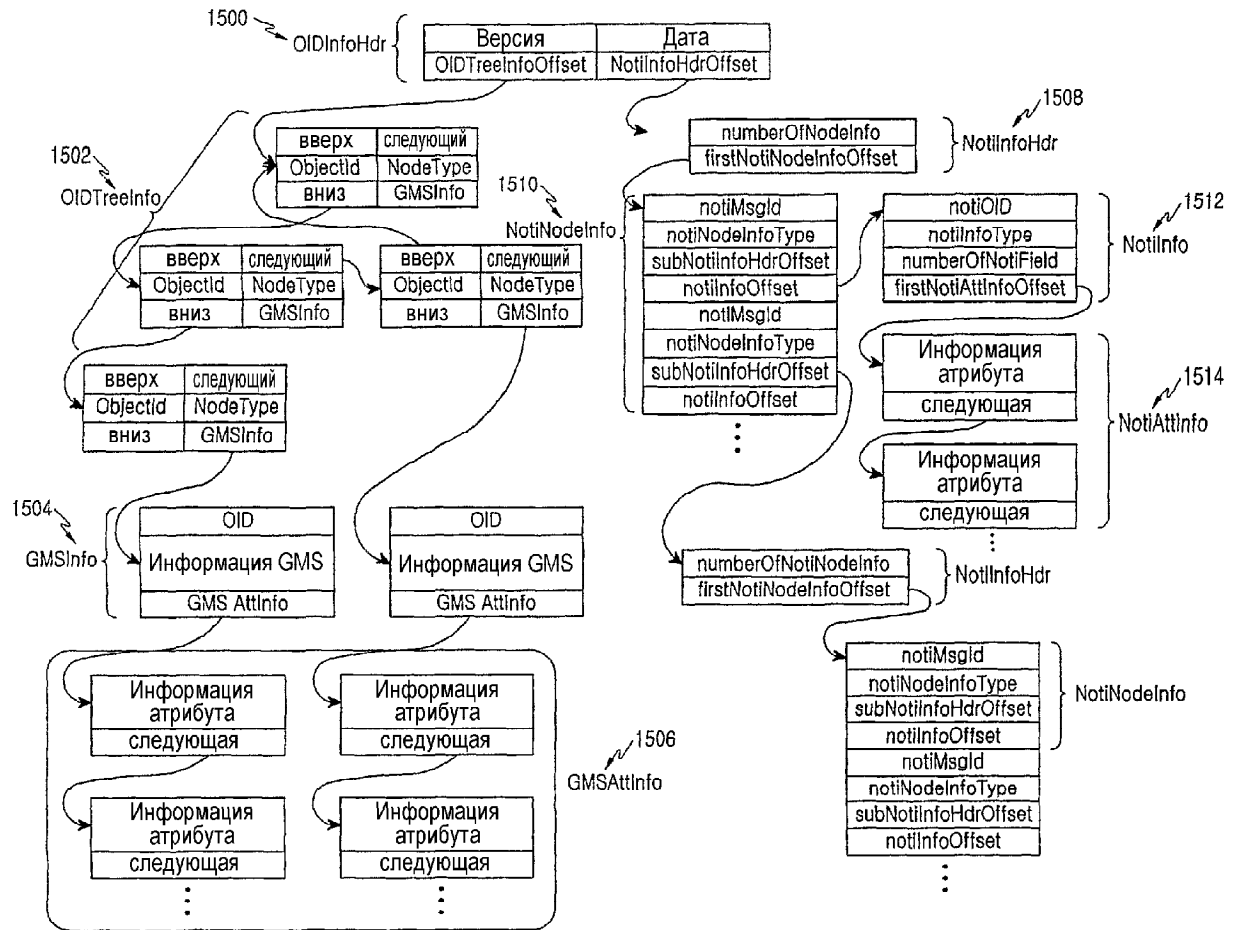
Фиг. 14



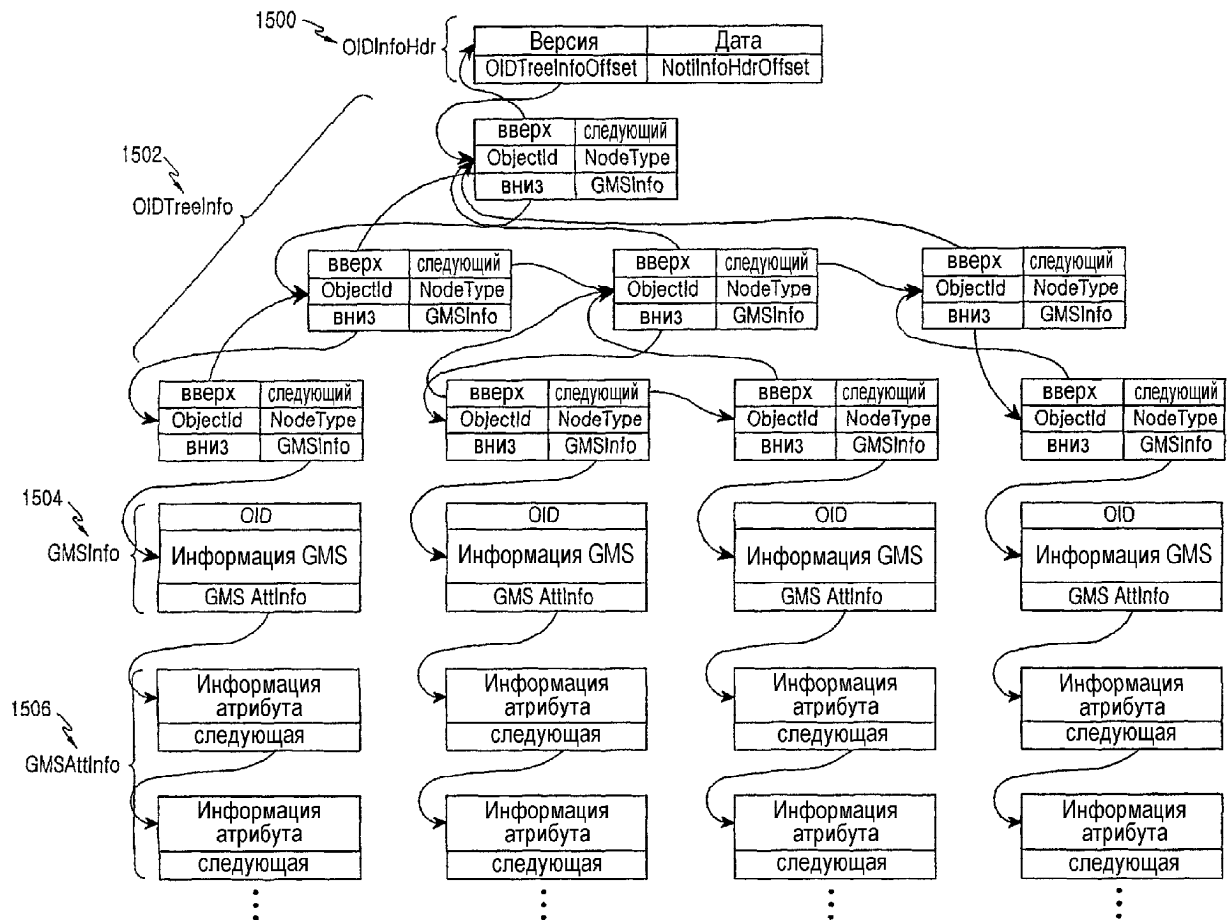
Фиг. 15А

Версия			Дата			1500
DefaultOID			OIDTreeInfoOffset		NotiInfoHdrOffset	
objectId	nodeType	upOIDTreeInfoOffset	downOIDTreeInfoOffset	nextOIDTreeInfoOffset	gmsInfoOffset	1502
objectId	nodeType	upOIDTreeInfoOffset	downOIDTreeInfoOffset	nextOIDTreeInfoOffset	gmsInfoOffset	
⋮						1504
OID			structName			
structType	payloadType	requestMsgId	responseMsgId	portNumber	numberOfIndex	
numberOfField	payloadSize	pldRelationId	masterTableOffset	nextGMSInfoOffset	firstGMSAttInfoOffset	
OID			structName			
structType	payloadType	requestMsgId	responseMsgId	portNumber	numberOfIndex	
numberOfField	payloadSize	pldRelationId	masterTableOffset	nextGMSInfoOffset	firstGMSAttInfoOffset	
⋮						
objectId	attName		asnType	accessType	fieldType	1506
attType	attSize	startOffset	subTableOffset	maxDimensionValue	nextAttInfoOffset	
objectId	attName		asnType	accessType	fieldType	
attType	attSize	startOffset	subTableOffset	maxDimensionValue	nextAttInfoOffset	
⋮						1508
numberOfNotiNodeInfo			firstNotiNodeInfoOffset			
notiMsgId	notiNodeInfoType		subNotiInfoHdrOffset	notiInfoOffset		1510
notiMsgId	notiNodeInfoType		subNotiInfoHdrOffset	notiInfoOffset		
⋮						1508
numberOfNotiNodeInfo			firstNotiNodeInfoOffset			
notiMsgId	notiNodeInfoType		subNotiInfoHdrOffset	notiInfoOffset		1510
notiMsgId	notiNodeInfoType		subNotiInfoHdrOffset	notiInfoOffset		
⋮						1512
notiOID	notiInfoType		numberOfNotiField	firstNotiAttInfoOffset		
notiOID	notiInfoType		numberOfNotiField	firstNotiAttInfoOffset		
⋮						
notiAttOID			notiAttName			1514
notiAttSize	notiAttType	notiASNTType	notiFieldType	subNotiAttInfoOffset	nextNotiAttInfoOffset	
notiAttOID			notiAttName			
notiAttSize	notiAttType	notiASNTType	notiFieldType	subNotiAttInfoOffset	nextNotiAttInfoOffset	
⋮						

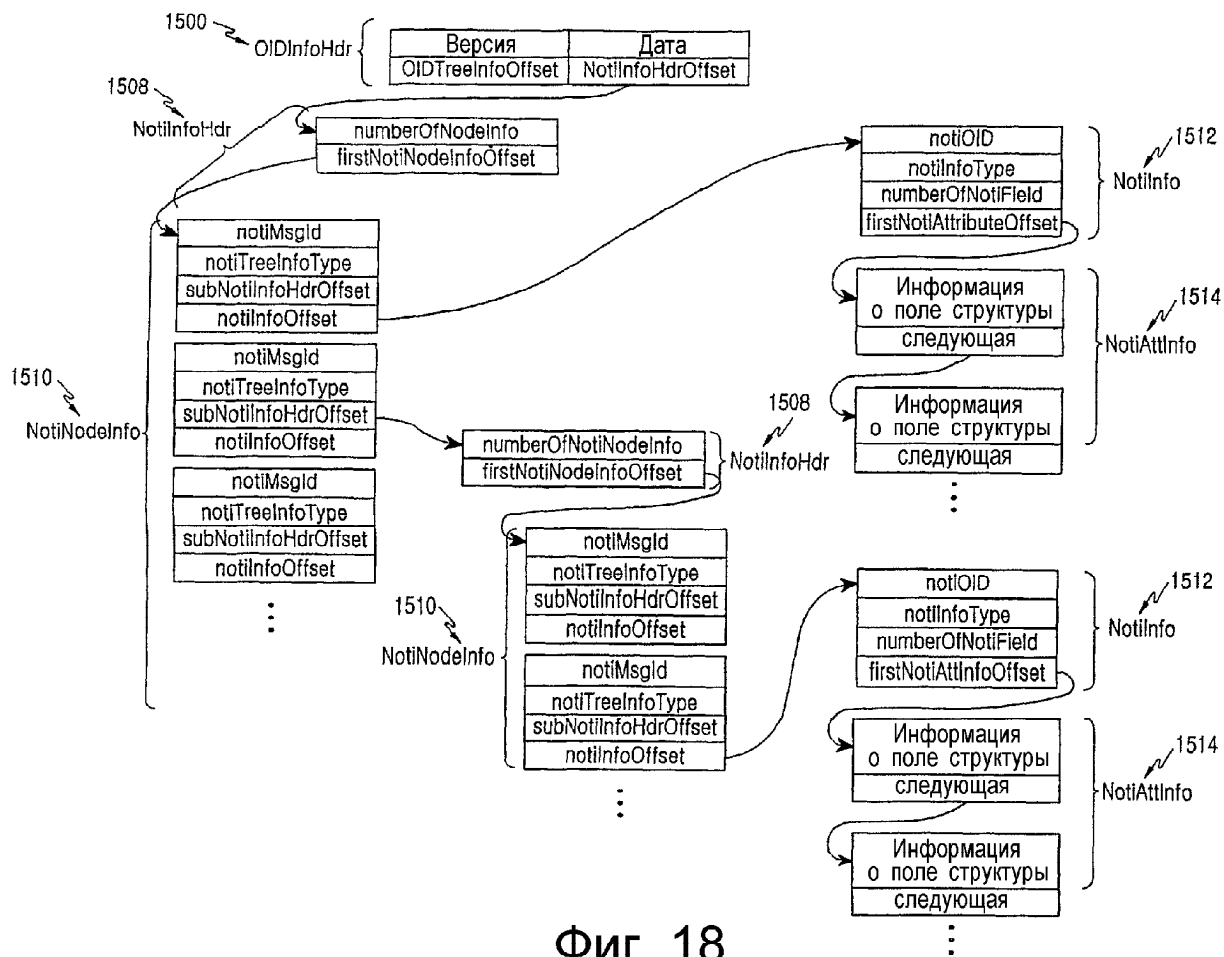
Фиг. 15Б



Фиг. 16



Фиг. 17



Фиг. 18