

DOI:

## УПРАВЛЕНИЕ ДИНАМИЧЕСКИМ ФОРМИРОВАНИЕМ КРУПНОМАСШТАБНЫХ СИСТЕМ ОБРАБОТКИ ИНФОРМАЦИИ С ИСПОЛЬЗОВАНИЕМ НЕЗАВИСИМЫХ БАЗ ДАННЫХ ЕДИНОЙ СТРУКТУРЫ

Орлов В.Л., Курако Е.А.

*Институт проблем управления им. В.А. Трапезникова РАН, Россия, г. Москва*

*ул. Профсоюзная д.65*

*ovl@ipu.ru, kea@ipu.ru*

*Аннотация: Рассматриваются способы обработки информации в крупномасштабных системах с использованием множества независимых баз данных, управляемых общим сервером приложений. Обсуждаются методы, основанные на использовании сервис-браузерной технологии. Предлагаются способы формирования баз данных единой структуры для обработки одной программной системой, ориентированной на глобальную базу данных.*

Ключевые слова: независимые базы данных, сервисы, множественная функциональная структура, системы, организации.

### Введение

Крупномасштабные системы представляют собой сложные системы, характеризующиеся разноплановым комплексным взаимодействием компонентов и территориальной распределенностью.

Обычно системы такого типа помимо развитого Центра содержат филиальные структуры (ФС), которые удалены друг от друга, но обладают функциональной схожестью. Исторически сложилось так, что создание и совершенствование ФС в определенный период проходило автономно. Но на этом этапе был важен единый интерфейс как между Центром системы и различными ФС, так и интерфейс взаимодействия ФС между собой (если в конкретной структуре такое взаимодействие требовалось). Вместе с тем конкретная реализация ФС при схожих функциях могла быть различной и даже разрабатываемой и выполняющейся на разных технологических платформах. Но единый интерфейс, хотя на его оттачивании и ломалось немало копий, был общим объединяющим моментом. По мере совершенствования методов связи, для обработки всех данных, хранящихся в разнородных источниках, были разработаны специальные технологии, основанные на Data Manning [1].

Другой этап развития крупномасштабных систем проходил под эгидой общих программных средств. При этом за основу выбиралась некоторая базовая система (БС), которая настраивалась, переделывалась, дополнялась новыми возможностями. Но, к сожалению, для разработчиков этого периода основной работой была БС, а идеальное внедрение – это настройка, даже сложная, но не требующая существенного изменения основных программных средств. Естественно, такие системы, первоначально разрабатываемые западными компаниями и под западные компании, были ориентированы во многом на несколько другой стиль управления, что вызывало трудности в процессе внедрения. Но зато ФС в этом случае, как правило, были однотипны, но обладали только определенной функциональностью, что вводило свои ограничения в работу.

Еще одним этапом стало использование web-технологий и применение браузеров в качестве клиентов [2]. Это, безусловно, шаг к централизации. Действительно, не нужно оборудовать клиентские места. На каждом рабочем месте используется системное программное обеспечение, присутствующее уже обычно в составе операционной системы. То есть для армии обслуживающего персонала – огромное облегчение. Не нужно на каждом компьютере устанавливать специальное программное обеспечение, разбираться с вопросами несовместимости оборудования и конкретного продукта, настраивать эти продукты, разбираться с неисправностями на месте, проводить периодические обновления. Все это не требуется, а с вопросами настройки общесистемного программного обеспечения разберутся местные администраторы общего профиля.

Но за все нужно платить. Все программное обеспечение сосредоточивается в Центре. Там концентрируются большие вычислительные мощности. Возникают мощные стойки оборудования, обеспечивающие распределение нагрузки, виртуализацию серверов, кластеризацию баз данных, гарантирующие непрерывный контроль и восстановление. И вслед за этим приходит целая армия обслуживающего персонала, без которого невозможно обойтись.

Филиальные системы, в этом случае, обычно размещаются в Центре. Они, как правило, также разрабатываются по единому шаблону. Нужно отметить, что в силу особенностей технологии обмен данными с клиентами, особенно по вопросам активного ввода информации, сокращен. При этом

браузеры функционируют на рабочих местах, а серверы базируются в центральной структуре, то есть нужно понимать, что передача информации в режиме «вопрос-ответ» преодолевает большие расстояния, но благодаря современному развитию сетевых технологий обмен данными осуществляется достаточно быстро.

Вместе с тем, по мере развития облачных технологий, появилась идея образования множественных функциональных структур, напоминающих филиалы, но являющихся независимыми образованиями. То есть, если классическая крупномасштабная система ориентирована на мощный центр с филиальной сетью, то крупномасштабная множественная структура организована так, что включает в себя много автономных информационных систем однотипной направленности [3]. И это во многом определяется тем, что каждая система, обслуживающая несколько организаций, выделяет для каждой из них независимую базу данных (БД) единой структуры.

## 1 Организация множественной функциональной структуры

### 1.1 Основные компоненты множественной функциональной структуры

Основные компоненты множественной функциональной структуры, будем называть ее в дальнейшем МФС, представлены на рис.1.

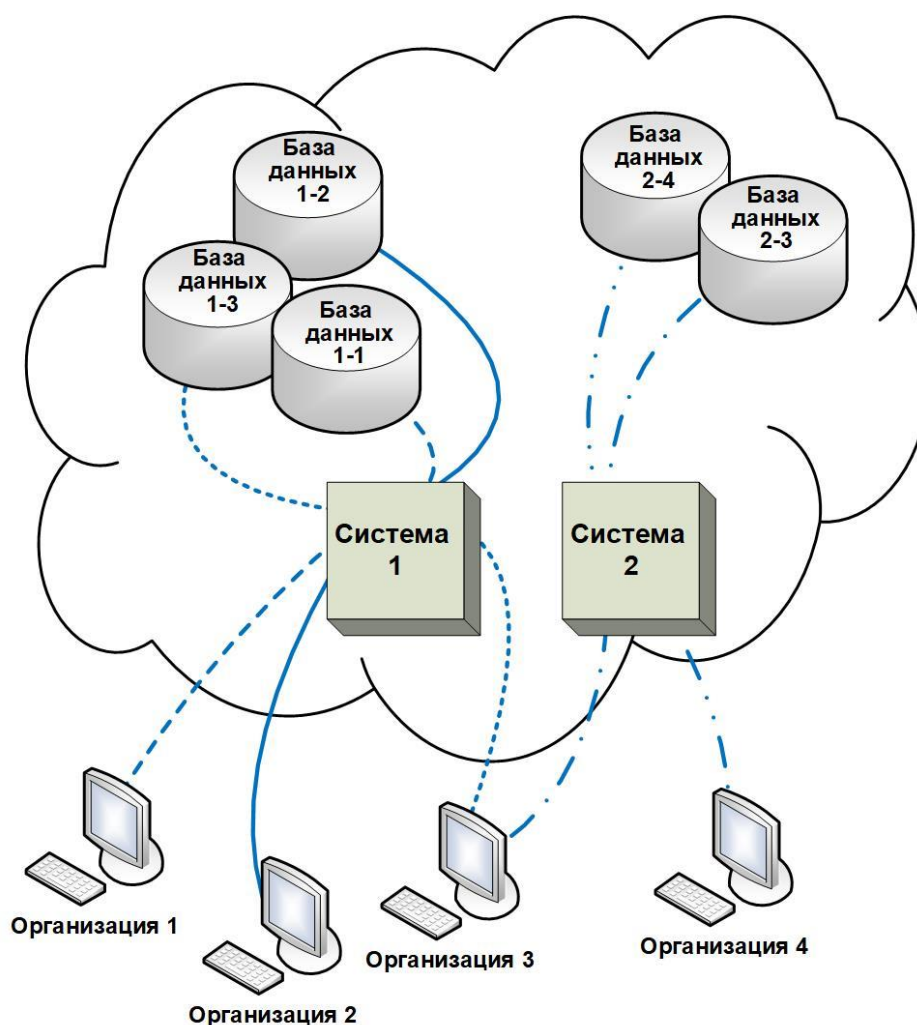


Рис. 1. Основные компоненты МФС

Основным компонентом МФС является информационная система. Причем система организуется таким образом, что содержит только программы и константы. Если к системе требуется подключить организацию, включающую множество рабочих мест, то для этой организации должна создаваться независимая база данных, структура которой будет совпадать со всеми базами данных, работающими с этой системой.

То есть структурно компоненты МФС содержат:

- Программы, обслуживающие рабочие места организации;

- Программы, реализующие ту или иную систему;
- Множество баз данных единой структуры для каждой системы.

В каждой из множества организаций  $\{Org\}$  может быть использовано множество рабочих мест  $\{Ws\}$ . Множество систем  $\{Sys\}$ , как правило, формируются за счет использования совокупности сервисов  $\{Srv\}$ , принадлежащих каждой системе.

А для каждой пары «Система-Организация» может быть создана своя база данных  $\{Db\}$ . При этом  $max\{Db\} = |Org| \times |Sys|$

То есть произведение мощностей организаций и систем дает максимальное значение количества создаваемых баз данных. Реальное количество баз данных может быть существенно меньше, так как не все организации связаны со всеми системами. Так в структуре, отображенной на рис.1, может быть создано восемь баз данных, а реально их построено пять, так как данная МФС не является полностью связанной по компонентам  $Sys-Org$  и реальные связи определяются потребностями организаций.

## 1.2 Управление МФС

Все основные компоненты МФС должны управляться. Действительно, на каждом рабочем месте, принадлежащем той или иной организации должна иметься возможность подключения к определенной системе или нескольким системам. Каждая система должна разрешать работу определенным пользователям. При подключении организации к системе должна создаваться новая база данных. Естественно, все эти управляющие функции не входят ни в одну из пользовательских систем.

Поэтому в МФС предусматривается управляющая структура, которая строится по тому же принципу, что и в пользовательских системах. То есть присутствует один или несколько сервисов, объединенных в управляющую систему, есть рабочее место организации, с которого начинается присоединение этой организации к одной из пользовательских систем и есть управляющая база данных (рис.2), в которой хранятся сведения, необходимые для управления другими прикладными системами и базами данных (БД).

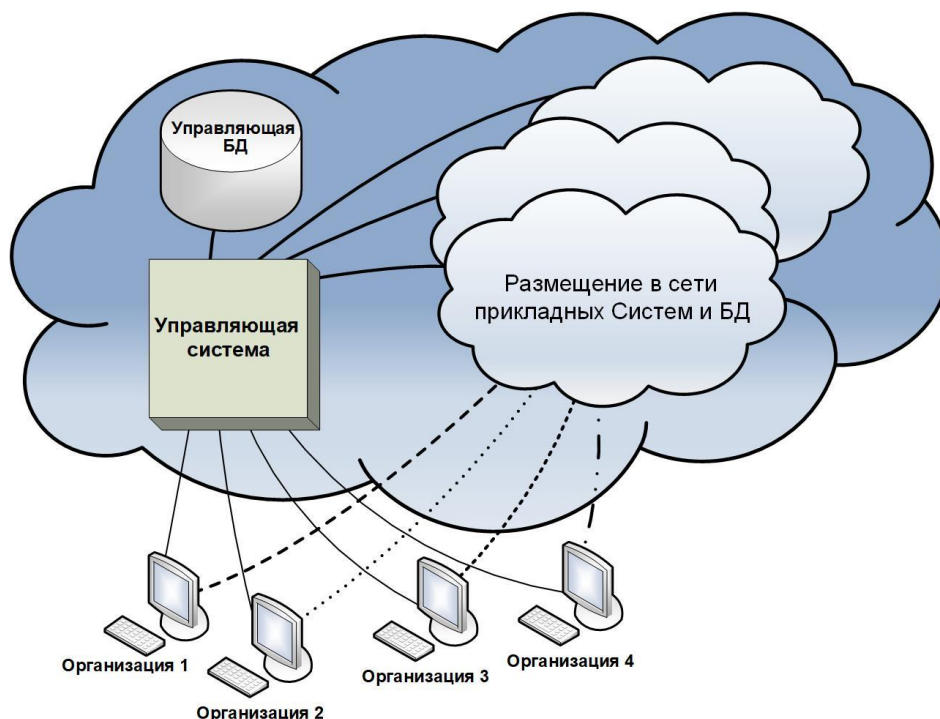


Рис. 2. Управление МФС

Администратор любой организации для подключения к МФС должен вызвать сервис управляющей системы. После диалога и проверок данные заносятся в управляющую БД. В случае успешного исхода, в сети отыскивается требуемая система и для пары «Система-Организация» формируется новая прикладная база данных. С одной стороны, прикладная БД имеет структуру, заданную выбранной системой. С другой стороны, в процессе диалога администратор организации предъявляет полномочия, в результате чего он получает доступ к прикладной БД и может с ней работать, вызывая функции системы.

Следует обратить внимание на следующее: при настройке МФС работа обязательно должна проходить как с управляющей системой, так и с прикладными системами. Это определяется тем, что конкретная система, работающая от имени зарегистрированной организации, использует вполне определенную БД. А это значит, что в процессе выполнения заданий пользователи организации обращаются как к определенным системам, так и к базам данных. Пример протокола начала работы представлен на рис.3.

Начало работы в МФС инициируется с рабочего места пользователя (клиента). При этом нужно иметь ввиду, что рабочих мест может быть несколько.

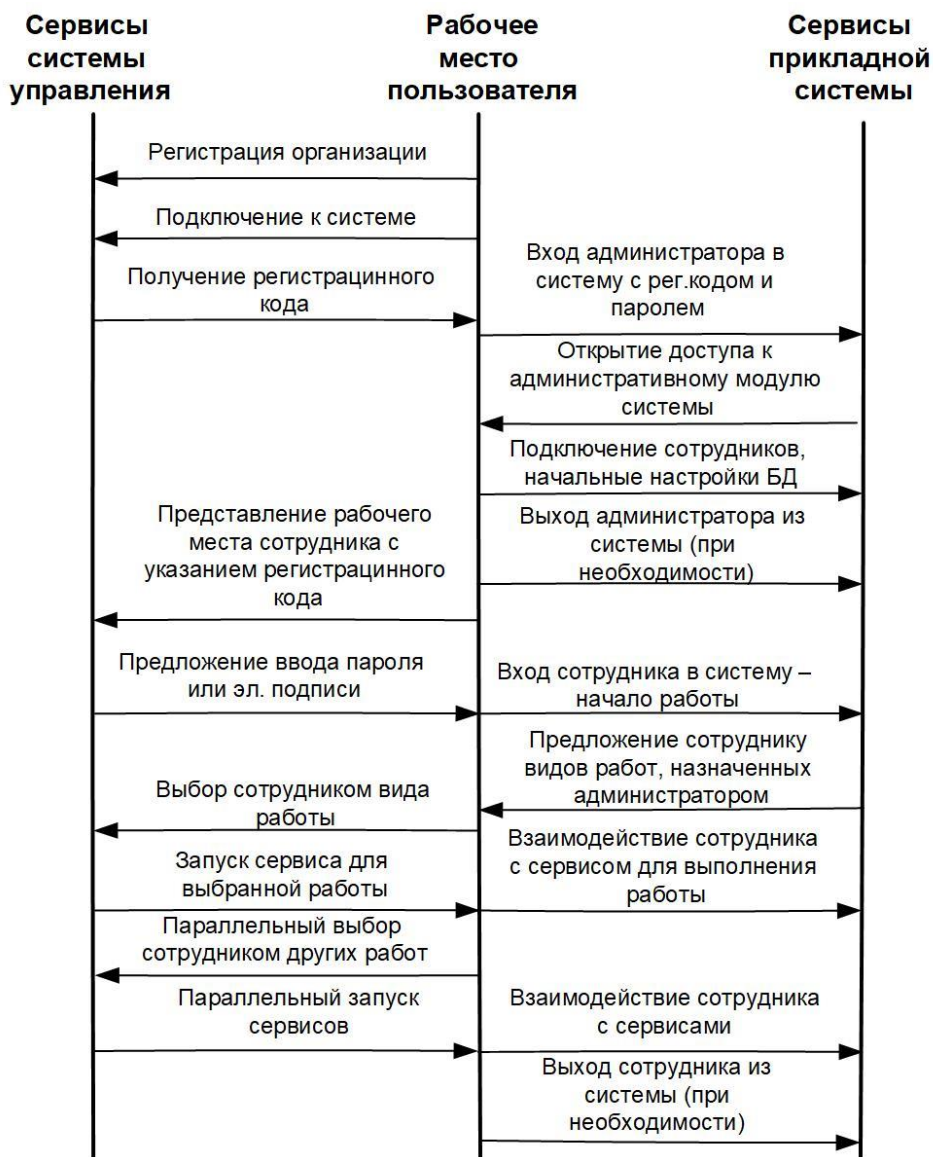


Рис. 3. Протокол начала работы в МФС

Обычно работа начинается с рабочего места администратора. При этом предполагается, что организация не имеет доступа к системам МФС и ее предварительно необходимо зарегистрировать. После регистрации организации происходит подключение к одной из прикладных систем при выполнении определенных условий, которые здесь не рассматриваются. Для пары «Система-Организация» создается уникальный регистрационный код, который по электронной почте или с использованием SMS отправляется пользователю-администратору.

В этом случае администратор может ввести полученный регистрационный код, а также логин и пароль, которые были зафиксированы в процессе регистрации. После этого администратор получает обычно доступ к административному модулю системы, где может провести предварительную настройку системы. В первую очередь формируется список сотрудников организации, которые имеют доступ к прикладной системе с указанием первичных данных для входа в эту систему, и список запускаемых работ для каждого сотрудника.

После этого администратор может выйти из системы, сообщить своим сотрудникам первичные входные данные, и каждый из них может получить доступ к системе со своего рабочего места, обеспечив ввод регистрационного кода, а также логина или пароля, предоставленными ему администратором организации. В последнем случае могут использоваться и другие методы, например, применение электронной подписи.

После входа в систему сотрудник обычно получает список разрешенных работ, к которым он может иметь доступ в системе. Обычно работа ассоциируется с некоторым сервисом, дающим возможность выполнить ее в режиме взаимодействия с оператором-сотрудником. Важно, что он может вызывать различные сервисы и параллельно производить необходимые операции.

### 1.3 Управляющие базы данных

Структура управляющих баз данных, использующихся в МФС, принципиально должна быть проста, так как выполняемые действия должны быть компактны и выполняться достаточно быстро.

Основные сущности, используемые в управляющей базе данных, представлены на рис. 4. Они представляют собой «Системы», «Организации» и «Список БД».

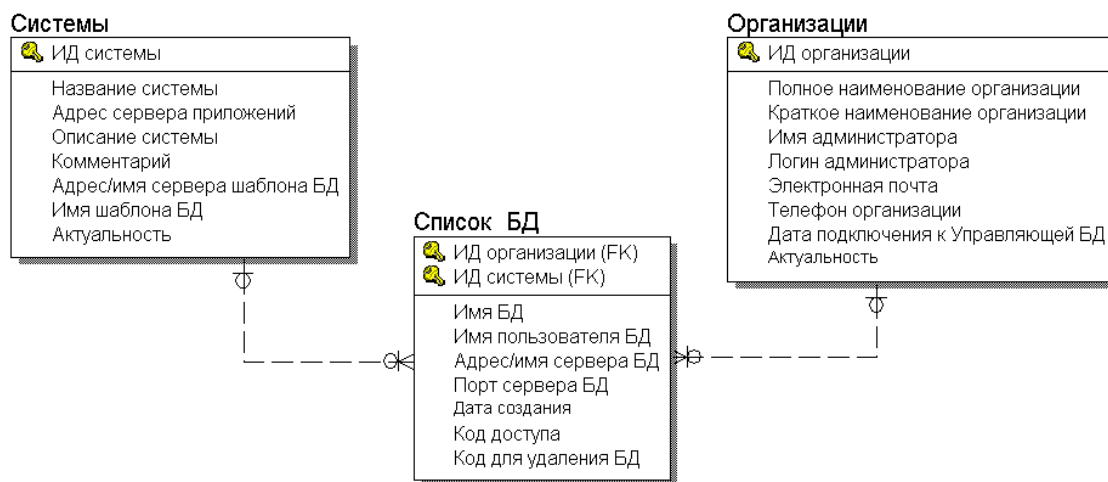


Рис. 4. Основные сущности управляющей БД

Сущность «Системы» кроме названия прикладной системы, адреса сервера приложений, где размещаются сервисы, а также описания, включает важные параметры, связанные с шаблоном БД. Это имя шаблона БД и адрес или имя сервера, где этот шаблон размещается. Понятие шаблона БД является важным для системы. Такой шаблон содержит образ базы данных, включающий все необходимые таблицы, связи и прочие атрибуты, причем таблицы основных справочников, имеющих общее значение – заполнены. Основная база данных для конкретной пары «Система-Организация» в начальный период времени по существу является копией этого шаблона.

Сущность «Организации» содержит описание организации и краткие данные администратора, причем обычно даже фамилия и инициалы не вводятся. После регистрации организации фиксируется дата подключения организации к управляющей БД. Отметим, что использование флажка «Актуальность» как, в сущности «Организации», так и в сущности «Системы» обычно ограничивает организацию в правах в спорных ситуациях, но сохраняет все данные в течение определенного периода с целью возвращения к нормальной работе.

Сущность «Список БД» связывает сущности «Системы» и «Организации» и комплексный ключ сущности, определяющей базы данных, является соединением ключей двух других сущностей. В этой сущности содержатся параметры, определяющие название, местоположение БД, «Код доступа», где хранится регистрационный код, и «Код для удаления БД», который должен также формироваться по заявке пользователя в серьезных случаях.

## 2 Организация защиты информации в МФС

Организации, которые активно могут работать в рамках МФС, по существу являются независимыми и даже не связанными друг с другом. В этом плане для каждой прикладной системы могут быть использованы свои средства обеспечения безопасности, отличные от других систем. Это



является безусловным достоинством. Но вместе с тем при обеспечении защиты в МФС должны применяться на начальной стадии некоторые общие моменты.

В частности, это касается ввода первичных данных администратора. Например, при регистрации в МФС вводится логин и пароль. При этом пароль не сохраняется, а сохраняется некоторый хеш [4], вычисленный по усложненному алгоритму, характерному для этого МФС. Таким образом, знание логина и пароля администратором и передача усложненного хеша в прикладную систему позволяет представителю организации первый раз осуществить вход. Вторым условием является организация доступа к сервису, позволяющему вводить данные других пользователей организации и проводить необходимые начальные настройки. Этих двух условий достаточно, чтобы можно было начать работать с любой системой, входящей в структуру МФС.

Вместе с тем, построение системы, как совокупности нескольких сервисов, позволяет реализовать и другой подход. Он заключается в том, что для разных систем могут использоваться единые средства защиты информации. Но эти средства работают с различными наборами данных, причем каждый из них включается в состав собственной системы и все наборы имеют одинаковую структуру.

Еще один существенный момент, связанный с безопасностью информации, хранящейся в БД. Важно, чтобы пароли всех баз данных МФС были различными и более того, чтобы сотрудник не мог узнать ни пароли смежных баз данных, ни пароль собственной БД. Это может быть достигнуто тем, что пароли вычисляются как символьные хеш-последовательности, вычисляемые на основе имен баз данных с соответствующими добавками по алгоритму, характерному для процедуры хеширования данной МФС. Причем пароли нигде не хранятся, а автоматически вычисляются при необходимости открытия БД. Это обеспечивает закрытость паролей для всех взаимодействующих с МФС сотрудников и невозможности организации доступа к базе данных способами, отличными от тех, которые предоставляет система.

### 3 Взаимодействие программ МФС на основе сервис-браузерной архитектуры

Для построения МФС могут быть использованы различные средства. Но в то же время использование сервис-браузерной архитектуры [5,6] представляется наиболее естественным. Это определяется тем, что:

- средства защиты информации встроены в сервис-браузерную архитектуру, что упрощает ее использование;
- для построения систем применяются сервисы без визуальной составляющей, что дает возможность встраивать их в системы различного назначения;
- модули могут при необходимости подгружаться на рабочие места, что позволяет выполнять на клиентской части достаточно сложные вычисления, разгружая при этом серверную часть.

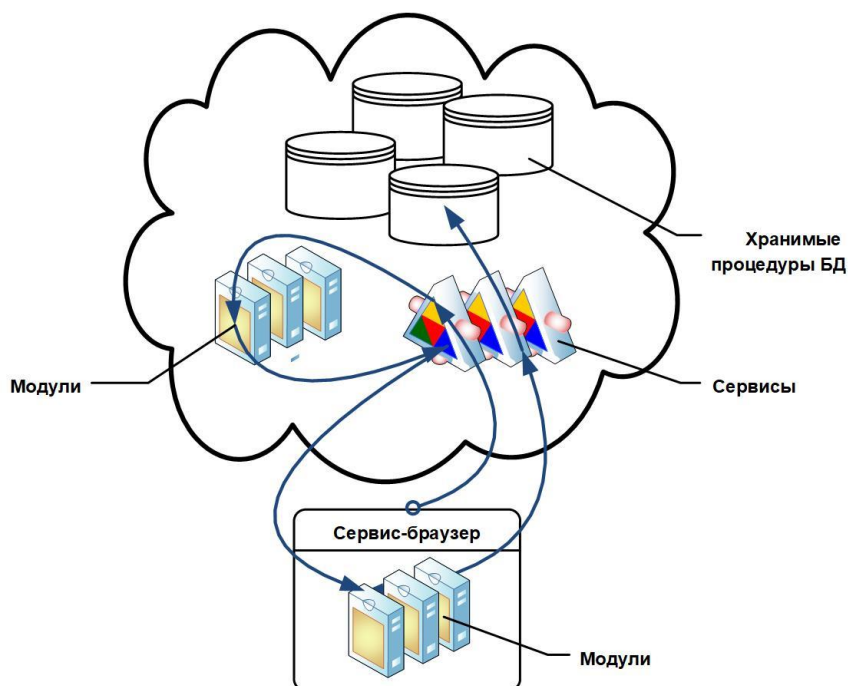


Рис. 5. Взаимодействие программ в сервис-браузерной архитектуре

При использовании сервис-браузерной архитектуры управляющие сервисы выбирают модули, хранящиеся на сервере, и копируют их на рабочее место пользователя (или обновляют), в свою очередь модули обращаются к прикладным сервисам, которые затем вызывают хранимые процедуры, входящие в состав БД.

Здесь стоит отметить несколько особенностей.

Во-первых, сервер-браузер после инсталляции не связан ни с одной из систем и не содержит прикладных модулей. Модули копируются в зависимости от выбора прикладной системы оператором. Если в дальнейшем модули не меняются, то в таком копировании нет необходимости. Производится копирование только тех модулей, которые изменились на сервере, то есть в этом случае осуществляется процедура обновления.

Во-вторых, для каждой системы имеется единственный набор серверов, которые работают с множеством баз данных единой структуры, созданных для каждой заявленной пары «Система-Организация».

В-третьих, в силу выбранной архитектуры вычислительная нагрузка равномерно распределяется между модулями рабочих мест, сервисами и хранимыми процедурами.

## Заключение

При использовании облачной технологии целесообразно создание в ее среде множественных функциональных структур, которые включают в свой состав одну или несколько систем, взаимодействующих с множеством независимых баз данных, причем для каждой системы база данных имеет единое строение. Несмотря на сложное внутреннее устройство и непростые действующие механизмы такие структуры удобны для пользователя, так как избавляют его от сложных процессов установки, сопровождения, обновления. То есть пользователь здесь работает по принципу: «подключился и работай».

Но чтобы соответствовать этому принципу, необходимо решить проблемы правильного выбора компонентов, управления структурой в целом, динамического и быстрого создания баз данных при подключении пользователей, обеспечения защиты информации и организации взаимодействия программ. Основные методы решения указанных проблем представлены в настоящей работе. Их практическое использование позволяет строить сложные системы обработки информации с использованием независимых баз данных единой структуры.

## Литература

1. *Ruili Wang, Wanting Ji, Mingzhe Liu, Xun Wang, Jian Weng, Song Deng, Suying Gao, Chang-an Yuan* Review on mining data from multiple data sources // *Pattern Recognition Letters*. Vol. 109, 2018. P. 120-128.
2. *Klaus-Dieter Schewe, Bernhard Thalheim* Design and Development of Web Information Systems. – Berlin, Heidelberg: Springer, 2019. – P. 599.
3. *Lebedev V.N. Kurako E.A. Orlov V.L.* Tasks of Small Business and Technologies of Large-scale Information Systems / *Proceedings of the 11th International Conference "Management of Large-Scale System Development" (MLSD)*. Moscow: IEEE, 2018. С. <https://ieeexplore.ieee.org/document/8551795>.
4. *Козлов А.Д., Орлов В.Л.* Методы и средства обеспечения информационной безопасности распределенных корпоративных систем. – М. ИПУ РАН, 2018. -155с.
5. *Курако Е.А., Орлов В.Л.* Сервис-браузеры для информационных систем // *Программная инженерия*. М., 2017. Т. 8, № 9. С. 413-421.
6. *Kurako E.A. Orlov V.L.* Service-browser Architecture and Large-scale Information Systems / *Proceedings of the 11th International Conference "Management of Large-Scale System Development" (MLSD)*. Moscow: IEEE, 2018. С. <https://ieeexplore.ieee.org/document/8551831>.