

УДК 681.3

DOI: <http://doi.org/10.20535/0203-3771372019177782>**В. В. Борисов**¹, старший преподаватель

АНАЛИЗ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ УПРАВЛЕНИЯ ПРОЕКТНЫМИ ДАННЫМИ

En

Structure stresses and deformations calculation veracity is determined by the modeling quality depending on how the structure of the design model and the elastic properties of its elements correspond to the structure and elastic properties of the strength elements of the simulated structure. As a result, the complexity of the procedure for the initial data setting (the analysis model), which becomes comparable with the complexity of the design itself in the case of a complex technical object designing, increases significantly. Taking into account the experience of modern computer-aided design systems using, the problem of reducing the computational models synthesis duration of the complex technical objects can be solved using the decomposition method. In this case, there is a problem associated with the need to harmonize the models structures of substructures, without which it is impossible to synthesize a common model of the structure. The process of structures coordinating of many models is associated with the intensive data exchange within the information system. In this case, the size and the structure of traffic substantially depend on the features of the interacting models structures implying a wide variety of data transfer formats.

Based on the results of the research, it has been established that canonical data structures supported by the majority of modern DBMS do not allow to effectively managing the process of data exchange between CAE complexes. It is shown that DBMS, using object-oriented information technologies, are the most optimal ones for managing the process of the computational models synthesis of the complex technical objects. At the same time, it is necessary to implement information technology, considered the objects as the active elements operating outside the program modules, to provide the full implementation of data exchange control functions between analysis models.

Ua

У статті аналізуються проблеми інформаційних технологій, що використовуються у сучасних розрахункових програмних комплексах, призначених для вирішення задач проектування складних технічних об'єктів. Досліджено схеми взаємодії розрахункових моделей елементів складних технічних об'єктів, а також проблеми використання сучасних СКБД для керування процесом обміну даними у межах проекту. Показано, що для керування процесом синтезу розрахункових моделей складних технічних об'єктів найбільш оптимальні об'єктно-орієнтовані СКБД.

Введение

Одним из наиболее важных, трудоемких и сложных видов расчета является статический расчет напряженно-деформированного состояния

¹ КПИ им. Игоря Сикорского

сборных конструкций. Он необходим для разработки структуры и определения оптимальных параметров элементов силового набора. Наиболее эффективным методом статического расчета конструкции является метод конечного элемента, позволяющий учитывать влияние упругих деформаций конструкций при оптимизации упругих свойств элементов силового набора, корректное задание всех видов нагрузок, а также граничных условий.

Необходимость сокращения сроков проектирования накладывает существенные ограничения на время выполнения расчетных работ, а значит, на время, необходимое для синтеза моделей. На основании анализа статистических данных, имеющих в свободном доступе, длительность цикла проектирования самолета транспортной категории, в среднем, составляет 6 лет (рис. 1).

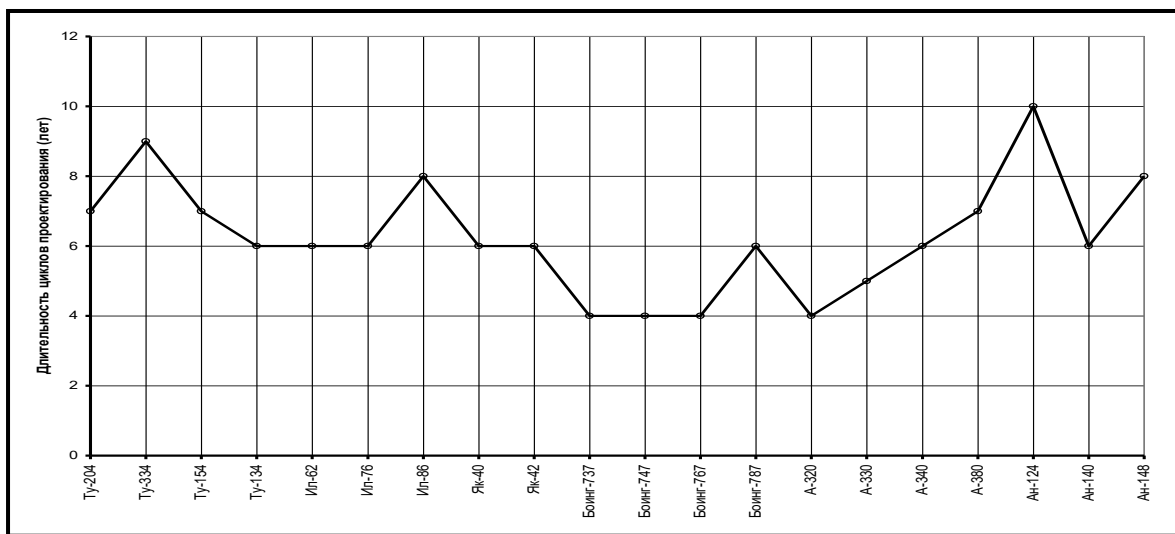


Рис. 1. Длительность циклов проектирования транспортных самолетов

В статье анализируются проблемы информационных технологий, используемых в современных расчетных программных комплексах, предназначенных для решения задач проектирования сложных технических объектов. Исследованы схемы взаимодействия расчетных моделей элементов сложных технических объектов, а также проблемы использования современных систем управления базами данных (СУБД) для управления процессом обмена данными внутри проекта. На основании результатов исследований установлено, что канонические структуры данных, поддерживаемые большинством современных СУБД, не позволяют эффективно управлять процессом обмена данными между расчетными программными комплексами (САЕ-комплексами). Показано, что для управления процессом синтеза расчетных моделей сложных технических объектов наиболее оптимальны СУБД, использующие объектно-ориентированные информационные технологии.

Анализ проблемы. При сокращении длительности проектирования, например, до 5 лет, максимальная длительность стадии оптимизации конструкции планера, соответственно, не должна превышать 2 года, или 4000 рабочих часов. Основой базового метода синтеза конечно-элементной модели (КЭМ), который применяется в современных САЕ-комплексах, наиболее часто используемых при выполнении проектного анализа прочности конструкций сложных технических объектов (СТО), является принцип автоматического разбиения на конечные элементы (КЭ) предварительно заданных областей. Как правило, такие области задаются в виде системы геометрических моделей (ГМ), каждая из которых описывает пространственное расположение и форму какого-либо фрагмента силового набора конструкции. На основании анализа существующих конструкций установлено, что среднее количество таких ГМ в кессоне крыла среднего самолета транспортной категории составляет примерно 15000. Формирование ГМ осуществляется, в основном, в интерактивном режиме. Если предположить, что на формирование одного фрагмента ГМ, с учетом анализа проектных данных, затрачивается примерно 5 минут, то для описания всей ГМ кессона крыла, при должном качестве моделирования, требуется 1250 рабочих часов. В результате, с учетом времени, необходимого для описания свойств КЭ и анализа результатов, за время проектирования может быть выполнен анализ не более одного варианта конструкции кессона крыла, что не обеспечивает необходимое качество проектирования.

Анализ нерешенных вопросов. С учетом опыта использования современных систем автоматизированного проектирования, проблема сокращения длительности синтеза КЭМ СТО может быть решена с помощью метода декомпозиции, в соответствии с которым единый процесс формирования общей ГМ конструкции преобразуется в систему взаимосвязанных процессов формирования ГМ подконструкций. Благодаря меньшему количеству элементов, ГМ подконструкций требует меньше времени для формирования. В результате появляется возможность, за счет параллельного формирования частных моделей, сократить общую длительность синтеза КЭМ. Однако при этом возникает проблема согласования структур ГМ подконструкций, без чего невозможно осуществить процедуру их объединения в общую ГМ конструкции. Процесс согласования структур множества моделей связан с интенсивным обменом данными внутри информационной системы. При этом объем и структура трафика существенно зависят от особенностей структур взаимодействующих моделей, что подразумевает большое разнообразие форматов передаваемых данных. Такой режим обмена данными может быть реализован только с помощью многопользовательской СУБД.

Проведенный анализ функций современных САЕ-комплексов, таких, как MSC.Nastran и ANSYS, показал, что ни один из них не имеет интерфейса с СУБД, а также не поддерживает функций СУБД [1,2], а значит не

обеспечивает необходимой производительности процесса моделирования конструкции планера СТО.

В отличие от описанного выше базового метода синтеза расчетных моделей, существует метод, обеспечивающий автоматическое формирование структуры КЭМ без использования предварительно сформированной ГМ [3]. При его разработке был впервые использован новый принцип декомпозиции КЭМ, в соответствии с которым общая КЭМ конструкции ТО последовательно делится на частные КЭМ, каждая из которых соответствует элементу конструкции, имеющему определенное функциональное назначение. Такой принцип декомпозиции предложено называть «функциональным». Использование функционального принципа декомпозиции позволило решить следующие задачи:

- автоматизация формирования структур частных КЭМ элементов конструкции без использования ГМ;
- автоматизация процесса объединения частных КЭМ в общую КЭМ конструкции СТО в целом.

Постановка задачи

Целью данного исследования является анализ функциональных особенностей современных СУБД, с целью определения наиболее оптимальных методов управления данными в системах автоматизированного проектирования СТО.

Особенности процесса управления проектными данными

В монографии [3] показано, что реализация технологии автоматизированного синтеза КЭМ СТО возможна только с помощью информационной технологии (ИТ), обеспечивающей поддержку следующих функций:

- поддержка прямого обмена данными между проектными моделями, без участия пользователей информационной системы;
- возможность межмодельного обмена данными любого формата;
- поддержка индивидуального пользовательского интерфейса для каждого класса КЭМ;
- возможность использования любого необходимого количества классов моделей, с возможностью определения новых классов КЭМ непосредственно в процессе формирования КЭМ СТО.

Кроме того, выбираемая ИТ должна обеспечивать реализацию алгоритмов синтеза структур моделей на всех уровнях декомпозиции КЭМ. Для объединения локальных КЭМ в общую модель (синтеза структуры КЭМ) требуется информация об узлах, подлежащих объединению. Без участия пользователей такая информация может быть получена только путем обмена данными с помощью алгоритмов, связанных конкретными ти-

пами структур данных, учитывающих особенности функционирования моделируемых конструкций, что предполагает высокую степень специализации программного обеспечения (ПО). В результате количество вариантов ПО, необходимых для синтеза КЭМ СТО, становится сопоставимым с количеством моделей в БД проекта.

Особенности информационных технологий *PDM*–систем

PDM–системы служат для управления информацией об изделии и проектными данными на протяжении всего жизненного цикла изделия [4]. Структура ПО большинства *PDM*–систем включает в себя ядро и сервисные программы, например визуализаторы. Кроме того, предусматривается подключение прикладного ПО, например модулей, предназначенных для построения проектных моделей.

Все ПО *PDM*–системы реализовано в виде стандартных программных файлов и не входит в логическую структуру данных проекта. Наращивание функциональных возможностей *PDM*–системы возможно двумя способами: путем модернизации ядра или путем подключения дополнительных программ. Такая ИТ не обеспечивает динамического подключения алгоритмов анализа и обработки данных. Поэтому перечень функций *PDM*–системы ограничивается функциональными возможностями ее ядра, сервисного ПО и предварительно подключенных прикладных программ.

Все современные *PDM*–системы поддерживают функции декомпозиции и синтеза проектных данных, чем обеспечивается необходимая производительность интерактивных технологий ввода данных, которые широко используются при формировании 3D–моделей конструкций СТО [4÷6]. Проектные данные в *PDM*–системах представляются в виде упорядоченных иерархических структур данных, состоящих из двух основных типов объектов – сборок и деталей. При этом формирование 3D–моделей осуществляется только на уровне деталей. Синтез 3D–моделей сборок осуществляется автоматически путем объединения всех 3D–моделей деталей, на которые ссылается сборка (синтез геометрии). Синтез геометрии моделей сборок осуществляется методом пространственного совмещения, не требующего обмена данными между соединяемыми моделями и корректировки содержимого их внутренних структур. Такая ИТ дает возможность реализовать декомпозицию и синтез проектных моделей на основе ограниченного числа универсальных алгоритмов, содержащихся в ядре *PDM*–системы и сервисном ПО.

С точки зрения теории БД, структуры данных *PDM*–систем представляют собой упорядоченные «деревья» [6], в которых детали (*parts*) выполняют функции «листьев», а сборки (*assemblies*) – функции внутренних узлов. Главным признаком древовидных структур данных является то, что

типы связей от «родителя» к «потомку» могут быть «один к одному» либо «один ко многим», а от «потомка» к «родителю» возможны только связи типа «один к одному». То есть, ни один узел «дерева» не может иметь более одного «родителя». Также исключены горизонтальные связи между узлами одного уровня. Решение этой проблемы возможно только путем введения «избыточности» в структуру «дерева» (рис. 2):

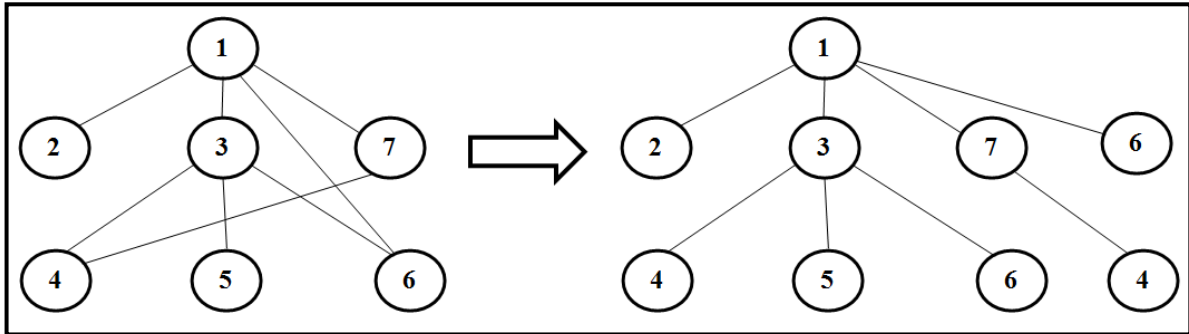


Рис. 2. Введение «избыточности» в структуру данных

Однако, данный способ подразумевает дублирование данных, которые не изменяются. Если дублированные данные меняются, возникает дополнительная проблема синхронизации их значений, для решения которой все равно потребуется поддерживать связи между дублируемыми данными. Указанные ограничения не позволяют реализовать схему обмена данными в рамках древовидной структуры данных. Теоретически, требуемая схема обмена данными может быть реализована в рамках сетевой структуры данных. Однако, в этом случае возникает проблема управления данными, которая не может быть решена в рамках информационных технологий, используемых в *PDM*-системах, поскольку для поиска уже существующих узлов необходима автоматическая генерация ключей, которая может быть обеспечена только с помощью большого количества специализированных программных модулей (ПМ). Следует также отметить жесткость организации данных в древовидных и сетевых структурах. Доступ к информации осуществляется только в соответствии со связями, определенными при проектировании структуры конкретной БД. БД с такими структурами сложно реорганизовывать. Их недостатком также является и сложность механизма доступа к данным, необходимость на физическом уровне четко определять связи данных. А поскольку каждый элемент данных должен содержать ссылки на некоторые другие элементы, то для этого требуются значительные ресурсы памяти ЭВМ. Для таких моделей характерна сложность реализации систем управления базами данных. Все это противоречит требованию возможности ввода новых типов КЭМ, непосредственно в процессе формирования структуры данных КЭМ.

Таким образом, на основании проведенного анализа можно сделать следующие выводы:

- *PDM*–системы не могут обеспечивать обмен данными между КЭМ без участия пользователей информационной системы.
- *PDM*–системы не могут поддерживать индивидуальный пользовательский интерфейс для каждого типа КЭМ.
- *PDM*–системы не позволяют использовать любое необходимое количество классов КЭМ, с возможностью ввода новых классов непосредственно в процессе синтеза КЭМ кессона крыла.

Анализ функциональных особенностей реляционных СУБД

Основные *PDM*–системы базируются сегодня на реляционных СУБД, лидирующей среди которых является продукт фирмы *Oracle*. Тип базовой СУБД определяет структуру и свойства ПО, которое может быть использовано *PDM*–системой для анализа и обработки данных, а также методы и средства управления данными. Реляционная модель данных не допускает естественного представления данных со сложной структурой, например КЭМ, поскольку в ее рамках возможно моделирование лишь с помощью плоских отношений (таблиц). Все отношения принадлежат одному уровню, многие значимые связи между данными либо теряются, либо их поддержку приходится осуществлять в рамках конкретного прикладного ПО. В составе *PDM*–систем реляционные СУБД в основном обеспечивают регистрацию моделей и формирование выборок на основании заданных атрибутов. Кроме того, реляционная СУБД обеспечивает управление доступом к данным. Структура данных, поддерживаемая реляционной СУБД, оптимизирована для выборки данных из одной или нескольких двумерных таблиц, для чего используются операции реляционной алгебры.

Для активизации этих операций используется специальный язык запросов – *SQL*, предназначенный для описания условий формирования «выборок». Таким образом, реляционный принцип представления данных, поддерживает два типа связей (рис. 3): постоянные связи между исходными отношениями, описываемые в процессе проектирования БД, и временные связи между исходными отношениями и «выборками». Недостатком такой схемы обмена данными является то, что система *SQL*–запросов требует описания алгоритмов поиска при каждом обращении к БД. Использование реляционной алгебры означает, что формат передаваемых данных описывается вновь при каждом запросе.

При этом возможны только прямые запросы, что делает невозможными последовательные запросы от модели к модели. В отличие от КЭМ, «выборки» не являются постоянными элементами БД, а потому не могут использоваться для хранения моделей.

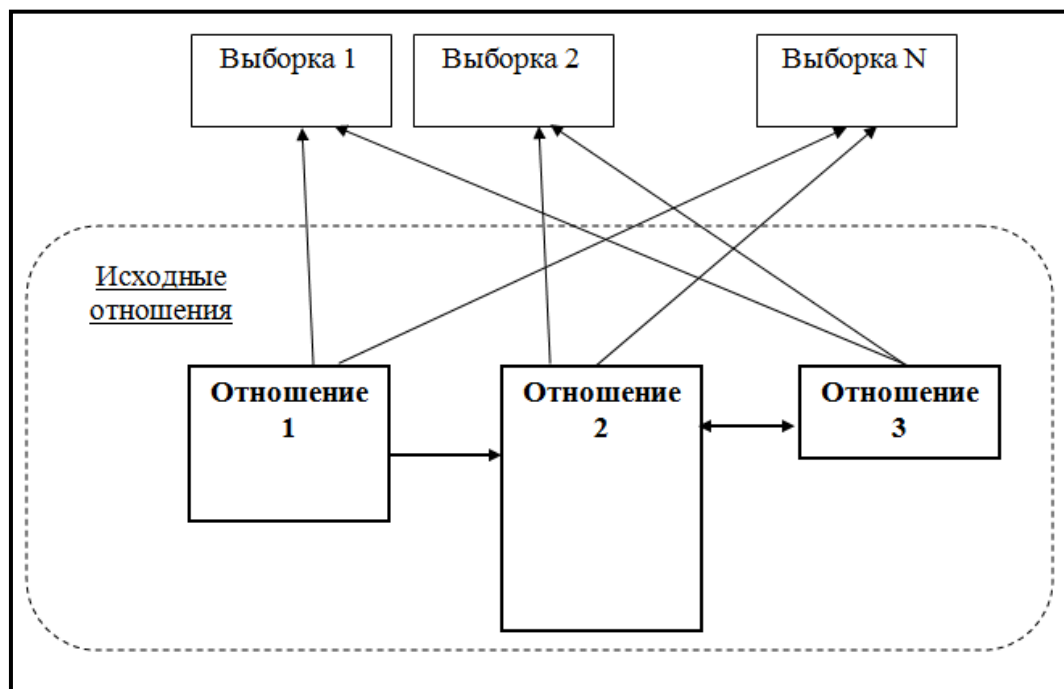


Рис. 3. Схема обмена данными в реляционной БД

Исходные отношения формируются вне процесса проектирования, под контролем лиц, отвечающих за целостность данных, и потому также не могут рассматриваться как проектные модели. Если КЭМ входят в структуру БД как данные, то есть хранятся в полях, предназначенных для произвольных данных, они рассматриваются реляционной СУБД как пассивные объекты, что не дает возможности обеспечить прямой обмен данными между ними.

Реляционная модель данных не предусматривает инкапсуляции данных на уровне концептуальной модели, используемой администраторами данных. В результате существенно затрудняется обмен данными между локальными информационными системами, входящими в состав САПР, поскольку для САПР характерен асинхронный режим обмена данными, при котором между приемом и передачей данных проходит довольно длительное время, в течение которого данные должны храниться во временных структурах данных. Включение таких структур в реляционную БД, существенно затрудняет процесс администрирования данных.

Важной особенностью реляционных СУБД является слабая приспособленность к расширению перечня типов данных, что противоречит условию о возможности использования любого необходимого количества классов КЭМ, с возможностью ввода новых классов непосредственно в процессе формирования КЭМ кессона крыла. Применение объектно-реляционных СУБД также не решает эту проблему, поскольку расширение списка типов данных в таких системах означает расширение ядра сервера, что невозможно делать в процессе проектирования [7].

Таким образом, на основании проведенного анализа, можно сделать следующие выводы:

- реляционная система управления данными не обеспечивает прямого обмена данными между КЭМ, в ней существенно затруднена процедура описания связей между моделями;
- слабая приспособленность реляционной СУБД к расширению перечня типов данных противоречит условию о возможности использования любого необходимого количества классов КЭМ, с возможностью ввода новых классов непосредственно в процессе формирования КЭМ кессона крыла.

Анализ функциональных особенностей объектных СУБД

Единственным типом современных СУБД, в которых реализован принцип инкапсуляции данных, являются объектные СУБД. В результате исследований объектных СУБД *IBM Lotus Notes/Domino*, *Jasmine*, *ObjectStore*, *Cache*, *Cerebrum*, *db4objects*, *Objectivity 5.0*, *ONTOS DB 2.5*, *Verasant*, *Release 5*, *Gemstone 5.0*, *POET 5.0*, *O2 5.0*, *Itasca 5.0*, *UniSQL 3.2* и *ODB–Jupiter 2.1* было установлено, что объектные СУБД обеспечивают более оптимальное решение задач управления проектными данными, поскольку объекты можно хранить и использовать непосредственно, не раскладывая их по таблицам. Кроме того, типы данных определяются разработчиком и не ограничены заранее определенным набором. Данные и методы объектов помещаются в хранилище как единое целое, чем, в частности, обеспечивается инкапсуляция. В результате, появляется возможность решить две проблемы обмена данными между КЭМ: обмен данными произвольного формата, определяемого требованиями прикладного ПО, и обеспечение временного хранения передаваемых данных без усложнения общей схемы данных.

Анализ показывает, что в соответствии с концепцией исследованных ОСУБД хранящиеся в БД объекты, а значит и хранящиеся в них модели, сами по себе не активны, поскольку их методы могут быть активизированы только в оперативной памяти ПМ. Например, ОСУБД *ObjectStore* обеспечивает долговременное хранение в базе данных объектов, созданных программами на языках *C++* и *Java*. Вся работа с объектами ведется обычными средствами включающего языка программирования. При этом объектная СУБД как бы расширяет виртуальную память операционной системы. На основании вышеизложенного, схему функционирования объектов в исследованных объектных СУБД можно представить следующим образом (рис. 4).

В объектной модели данных количество экземпляров алгоритмов, используемых для анализа и обработки данных, всегда соответствует количеству экземпляров структур данных, что, в принципе, позволяет решать

задачи, использующие большое количество алгоритмов. Анализ концептуальных моделей объектных БД показал, что все они формируются в виде хранилищ объектов, в которых функции объектной СУБД ограничиваются обеспечением целостности данных, а также поиском объектов в БД.

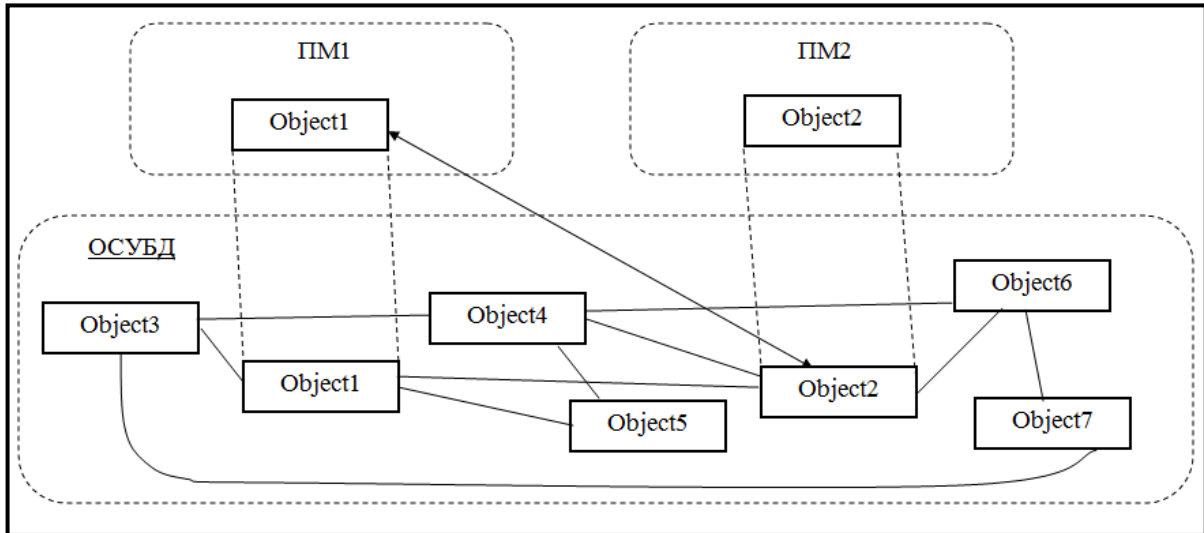


Рис. 4. Типовая схема функционирования объектной СУБД

Активизация методов объектов и изменение их свойств производится только в процессе функционирования ПМ, в оперативную память которых загружаются объекты (экземпляры классов). Рассматривая объекты как части прикладного ПО, исследованные объектные СУБД не обеспечивают автоматического обмена данными между моделями, хранящимися в объектах, так как невозможен прямой обмен данными между объектами, находящимися в разных модулях прикладного ПО. Кроме этого структура объектных БД не содержит связей между объектами. Связи между объектами описываются в исходных текстах прикладного ПО, только в процессе программирования, что не позволяет решать задачу синтеза КЭМ СТО.

Для решения этой проблемы предложена объектно-ориентированная ИТ, рассматривающая объекты как активные структурные элементы, функционирующие вне ПМ. При активизации объекты загружаются в оперативную память компьютера самостоятельно. Это позволяет компилировать класс объекта как отдельный программный код и вычислять исполнительные адреса структур данных на этапе компиляции, что дает возможность разработать унифицированную процедуру формирования объектов непосредственно в БД, без использования ПМ. Все данные, необходимые для формирования объектов, определяются в процессе компиляции классов и хранятся в специальной БД классов [8], откуда считываются при формировании объектов. Это дает возможность задавать класс объекта непосредственно в процессе его генерации, чем обеспечивается возможность определения классов расчетных моделей непосредственно в процессе их синтеза. Использование для внутриобъектного ПО системы адресации формата

Addr = Space:Obj:Seg:Offset позволяет реализовать особый метод обмена данными, названный "пересечением объектов", в соответствии с которым два и более объекта могут ссылаться на одну и ту же область памяти объектной БД. Для реализации этого метода в языке предложено использовать особые, "виртуальные", типы данных [8]. В отличие от обычных типов данных, виртуальные переменные содержат ссылки на области памяти в объектной БД. В отличие от обычных указателей, виртуальные переменные могут ссылаться на данные в любом объекте любой БД, находящейся под управлением объектной СУБД. Поэтому при их инициализации используется расширенный формат исполнительного адреса.

Выводы

На основании проведенного анализа особенностей ИТ, применяемых в современных СУБД, можно сделать следующие выводы:

- ИТ, используемые основными *PDM*-системами, не поддерживают функции, необходимые для синтеза КЭМ СТК.
- реляционные СУБД не обеспечивают автоматический обмен данными между моделями, обмен данными произвольного формата и поддержку индивидуального интерфейса для каждого типа моделей.
- объектные СУБД, наиболее полно отвечают задачам синтеза расчетных моделей конструкции СТК. При этом, для полной реализации функций управления обменом данными между расчетными моделями необходимо реализовать ИТ, рассматривающую объекты, как активные элементы, функционирующие вне ПМ.

Список использованной литературы

1. *MSC. Nastran 2012. Linear Static Analysis. User's Guide.* 2012. –С. 772.
2. www.ans.com.ru. Руководство по основным методам проведения анализа в программе ANSYS. 2006. –С. 399.
3. *Борисов В. В.* Методы синтеза конечно-элементной модели планера грузового самолета / В. В. Борисов. – Саарбрюккен: *LAP LAMBERT Academic Publishing*, 2014. – 139 с.
4. Windchill, ProjectLink. - <http://www/urss.ru/>.
5. Программный комплекс АРМ WinMachine. –<http://www.consistent.ru/download/marketing>.
6. *CATIA*, универсальная *CAD/CAM/CAE/PDM* система. –<http://www.catia.ru/productsNEW.htm>.
7. *Уорсли Д., Дрейк Д.* *PostgreSQL* для профессионалов. -М.: Питер, 2003. –С. 498.

8. *Борисов В. В.* Система автоматизированного управления проектными данными / В. В. Борисов, В. П. Зинченко, И. П. Муха// Адаптивные системы автоматизированного управления. –2011. №19(39). -С. 23–34.