

Единое электронное информационное пространство – универсальный инструмент построения сложных электронных моделей.

АННОТАЦИЯ

В данной статье представлено краткое описание архитектуры и основных принципов функционирования программного инструментария, предназначенного для построения распределенных в вычислительных сетях сложных электронных моделей. Дано определение **“Единого информационного пространства (ЕИП)”** как системы размещенных в узлах вычислительной сети взаимосвязанных идентичных программных модулей, функционирующих в режиме реального времени. Представлено описание основного элемента ЕИП – **программного узла ЕИП**. Показан пример построения средствами ЕИП типовой модели **“Электронное предприятие”**.

ВВЕДЕНИЕ

Анализ процесса развития мирового рынка программного обеспечения (ПО) с момента его возникновения и до сегодняшних дней позволяет выявить одну важную, хотя и вполне естественную, закономерность – развитие это происходит в соответствии с основными философскими законами. Сначала поступательное движение от автоматизации простых задач к все более сложным. Затем, в какой то момент, качественный скачок: реализация накопленных знаний в одном решении, базирующемся на новых платформах программирования. Затем вновь, поступательное движение и вновь – качественный скачок. Таким образом, процесс развития ПО происходит в соответствии с общемировыми законами эволюции. Из этого вытекает важный вывод: периодически в развитии ПО наступает момент, когда нет дальше другого пути, кроме как отбросить все старое и на основе накопленного опыта создать нечто новое, качественно иное. Особо следует отметить следующий факт: **никогда новый программный продукт не является простой суперпозицией старых!**

Современный этап развития программного обеспечения как раз находится в состоянии назревшего качественного скачка. В рамках глобальной компьютеризации автоматизированы практически все основные информационные и технологические процессы. Создан обширный парк специализированных программных комплексов, практически идеальных в рамках своих зон ответственности. Дальнейшее развитие ПО невозможно без постановки новых, более глобальных задач.

Сейчас цель **качественного скачка** - перейти от автоматизации отдельных информационных процессов к **автоматизации в целом жизненного цикла сложного административно-хозяйственного объекта**, в котором эти процессы протекают. Решить эту задачу старыми средствами невозможно просто потому, что существующее ПО создавалось для других задач, является уникальным и часто несовместимым между собой. И потому единственный и безальтернативный путь – **создать заново качественно иной программный продукт**, специально предназначенный для решения поставленной задачи.

1. Постановка задачи

Итак, нашей целью является создание программного инструментария, обеспечивающего построение электронных моделей сложных административно-хозяйственных образований (предприятия, научного учреждения, объекта административной структуры и т.п.). От обычных объектов указанные модели отличается одно важное свойство – **высокая степень изменчивости**. Мало того, что структура такого объекта не может быть заранее точно определена. Она еще должна меняться с течением времени под воздействием постоянно меняющегося внешнего окружения (изменение стандартов функционирования, новые технологии, новые нормативные данные и т.д.). При этом, электронная модель данного образования должна (находясь в распределенном состоянии) сохранять свою **информационную целостность**, обеспечивать **единообразие и единственность** представления данных, иметь **единый механизм генерации и перемещения данных**. И, наконец, моделируемые объекты могут настолько сильно отличаться по внутренней структуре, видам данных

и структуре информационных потоков, что загнать все это разнообразие в рамки какой-то одной глобальной модели не представляется возможным. Да и нужно ли это? В природе нет ни одного образования, претендующего на звание всеобъемлющего. Вместо этого мы видим огромное множество образований живой и неживой материи, представляющих собой самостоятельные физические объекты. Но созданы эти объекты на единой физической основе, функционируют в рамках единых физических, биологических и социальных законов. И при этом многие из них находятся в постоянном взаимодействии.

Отсюда сама собою напрашивается формулировка задачи: **нужно строить** не одну всеобъемлющую электронную модель, а **универсальный конструктор электронных моделей**. Физически данный конструктор должен представлять собой распределенную вычислительную среду, обладающую единой формой представления данных, единым механизмом формирования и взаимодействия создаваемых электронных моделей. Эту вычислительную среду мы и будем далее понимать под термином: **“Единое информационное пространство (ЕИП)”**. А поскольку данное образование тоже является, хоть и распределенным, но все-таки электронным объектом, оно может иметь и собственную модель представления данных, и все другие атрибуты электронного объекта.

Исходя из изложенных условий, можно в самых общих чертах представить себе архитектуру будущей электронной модели ЕИП:

- 1) Физически электронная модель должна быть исполнена в виде **компактного** многофункционального исполняемого **программного модуля** – узла ЕИП. Информационное пространство в целом есть совокупность идентичных узлов ЕИП, объединенных единой системой адресации. Только в таком исполнении возможно существовании ЕИП в распределенных информационных сетях.
- 2) Иметь универсальный способ представления данных любого вида, включая заранее неизвестные;
- 3) Использовать единый универсальный механизм создания, трансформации и перемещения данных, сведя к минимуму необходимость конвертирования данных при обмене между узлами ЕИП.

Очевидно, что никакая комбинация существующего ПО не удовлетворяет перечисленным требованиям. Поэтому **единственный путь решения** поставленной задачи – **построить заново** электронную модель основного элемента единого информационного пространства – **программного узла ЕИП**, максимально включив в нее весь известный функционал существующего ПО.

2. Формирование электронной модели ЕИП

2.1 Выбор общей архитектуры электронной модели.

В основу архитектуры электронной модели положена **древовидная иерархическая структура** представления данных. Основанием выбора явились свойства данного типа модели:

- **Простота**. Каждый элемент древовидной структуры может иметь одного ‘хозяина’ и множество ‘подчиненных’. Этих свойств достаточно для построения сколь угодно сложных вложенных структур данных;
- **Эргономичность**. Древовидная структура по сути своей является списочной конструкцией, а это самая компактная форма представления данных (в отличие, например, от табличной);
- **Гибкость**. Древовидную структуру можно трансформировать до неузнаваемости путем простой перестановки ветвей дерева, не оказывая при этом существенного влияния на другие участки структуры;

2.2 Состав основного элемента - программного узла ЕИП

Программный узел ЕИП представляет собой оформленный в виде единого исполняемого программного модуля набор независимых программных методов. Программные методы могут

взаимодействовать между собой, образуя разнообразные алгоритмические цепочки в соответствии со спецификой решаемой задачи. Фактически **программный узел ЕИП** можно рассматривать как многофункциональный **электронный объект-трансформер**, включающий в себя ограниченный набор базовых электронных объектов. Данные объекты могут быть использованы для построения древовидных структур данных, описывающих моделируемые физические, математические или информационные объекты. Эти же базовые объекты служат для формирования информационных процессов, функционирующих в рамках создаваемых математических моделей.

Составляющие элементы математической модели программного узла ЕИП можно условно классифицировать следующим образом:

- 1) **Базисный элемент** – обезличенный электронный объект-список, содержащий в своем списке в качестве подчиненных другие объекты-списки. Это основополагающий ‘кирпичик’ всей ЕИП. Все остальные электронные объекты (включая само ЕИП) являются производными от данного элемента;
- 2) **Компонент математической модели** – специализированный электронный объект-список, моделирующий конкретный физический объект. Например:
 - **Объект-микросхема** = список объектов-контактов + список графических примитивов для описания изображения;
 - **Объект-документ** = список объектов-страниц + список объектов-таблиц + список объектов-ячеек + список объектов-строк + список объектов-строк + список объектов-слов + список символов.
- 3) **Графический интерфейс** – список графических объектов управления (окно интерфейса, меню и кнопки управления, информационные панели и т.п.). В этом же списке содержатся компоненты модели, управляемые данным интерфейсом. Располагая в окне меню-интерфейса необходимые объекты управления с прикрепленными к ним соответствующими программными методами, мы можем построить специализированный интерфейс необходимого нам вида, выполняющий заданные нами виды действий над данными, помещенными нами же в список подчиненных данного интерфейса. К каждому графическому объекту управления может быть ‘прикреплен’ программный метод, запускаемый при активизации объекта управления.
- 4) **Информационный процесс** – специализированный объект управления потоками данных, базовый элемент для построения сложных информационных процессов. Данный элемент способен в режиме реального времени автоматически опрашивать указанные области файловой системы ПЭВМ и сети, выборочно считывать из них данные, анализировать считанные данные и сопутствующие им управляющие инструкции, передавать данные другим информационным процессам.

Следует отметить, что данная классификация достаточно условна. Все перечисленные элементы являются производными от единого базового элемента (1), и различаются между собой наборами используемых программных методов и внешним видом. Базисная структура и логика функционирования у всех объектов – идентичны.

2.3 Выбор базисного элемента хранения данных.

Любая электронная модель, независимо от объема и сложности, включает в себя две составных части. Это **данные** (составляющие тело модели) и **процессы** управления данными.

В основу универсальной системы представления данных должен быть положен универсальный же элемент представления данных. При создании универсального элемента представления данных можно пойти двумя прямо противоположными путями:

- а) **Путем усложнения.** Можно попытаться построить максимально полный базовый электронный объект представления данных, включающий в себя все возможные варианты данных и все программные методы управления. Идея заведомо порочная, хотя бы потому, что нам не дано предугадать будущее, и значит, всеобъемлющий базисный элемент невозможен в принципе.

б) Путем упрощения. В самом деле, основу мироздания составляют достаточно простые вещи! Это мы сами усложняем их понимание, пытаюсь на основе придуманных нами (часто – неверных) трактовок реальных физических законов описать окружающую реальность. Получается сложно и не точно. А между тем, Природа сама подсказывает правильный путь – **в основе более сложного всегда лежит более простое!** Достаточно взглянуть на периодическую таблицу Менделеева, чтобы увидеть, что в основе всего окружающего нас мира (живого и неживого) лежит **всего один элемент!** Это атом водорода! Все остальные элементы таблицы являются продуктами слияния различного количества ядер атома водорода. Молекулы веществ есть продукт объединения различных комбинаций атомов. Продвигаясь дальше по пути укрупнения, мы, в конце концов, дойдем и до ”венца творения” – человека разумного.

Таким образом, при правильном подходе, задача конструирования универсального элемента представления данных неожиданно существенно упрощается. Оказывается, **простота – залог универсальности!** Поэтому к базовому элементу хранения данных применимы те же критерии, что и при выборе архитектуры общей модели: раз вся модель – списочная, то и **базовый** ее ‘кирпичик’ тоже должен быть **объектом-списком!** Такой подход дает нам огромные преимущества, как при построении модели, так и при ее эксплуатации. Ведь если и макро- и микро-элементы модели однотипны, то для управления этими элементами можно построить **единый механизм управления.**

2.3.1 Свойства базисного элемента хранения данных.

Отдельный вопрос, каким образом достигается универсальность базисного электронного объекта и всей модели в целом? Ответ кроется в структуре и внутренних свойствах создаваемых электронных объектов. Дело в том, что электронная модель ЕИП строится с использованием методов объектно-ориентированного программирования и называется соответственно – объектная модель.

В объектной модели возможен только один тип используемых данных – электронный объект. Фактически объектная модель представляет собой набор своего рода ‘черных ящиков’, каждый из которых может иметь совершенно произвольное внутреннее содержание, но при взгляде извне, со стороны общей структуры модели и методов управления моделью – все ее элементы однотипны. Это означает, что в данной модели можно применить **единые методы** создания, редактирования, перемещения и хранения объектов. А это и есть главный признак универсальности модели.

Теперь рассмотрим, что собой представляет базовый элемент ЕИП и в чем его универсальность? Как уже ранее было упомянуто, это объект-список. Но это только самое поверхностное описание. На самом деле данный объект имеет довольно сложную структуру (так же как атом водорода не является минимально возможным физическим объектом). Просто для решения нашей прикладной задачи достаточно того набора внутренних свойств базового объекта, которые мы принимаем как данность. А свойства эти, вытекающие из реализации принципов объектно-ориентированного программирования, имеют ключевое значение для обеспечения универсальности базового элемента ЕИП:

- 1) Свойство **самодостаточности**. Классический электронный объект содержит в себе только данные собственного описания (является пассивным объектом). Все управление таким объектом осуществляется извне. Самодостаточный объект, кроме данных, содержит еще и набор программных методов, т.е. может самостоятельно управлять своими данными. Фактически каждый такой объект является самостоятельным исполняемым программным модулем, функционирующим по своим внутренним правилам, но с учетом внешних управляющих воздействий. При этом на долю внешнего управления остается самая незначительная часть функций. Как правило, это наиболее общие для всех объектов функции.
- 2) Свойство **наследования**. Новый самодостаточный объект может быть создан как некая производная от уже существующего объекта. В этом случае новый объект приобретает (наследует) все свойства и программные методы своего родителя (а так же родителей родителя). Это в сотни раз сокращает затраты на создание новых электронных объектов, поскольку, единожды реализованные для родителей свойства и методы автоматически передаются всем наследникам.

- 3) Свойство **единственности**. Это свойство вытекает из предыдущего и касается программных методов объекта. Заключается оно в том, что наследуемые программные методы (в отличие от данных) не дублируются физически в объектах-наследниках. Каждый метод при построении общего исполняемого модуля фиксируется в нем единожды. В дальнейшем, в процессе исполнения программы, каждый объект передает управление соответствующему программному методу и затем забирает результат. Данное свойство многократно сокращает размер общего исполняемого модуля модели ЕИП. Кроме того, существенно упрощается процесс отладки, поскольку внесенные изменения одновременно отображаются в работе всех компонентов модели.
- 4) Свойство **рекурсии**. Специфическое свойство, позволяющее программному методу циклически инициировать выполнение самого себя. Это мощнейшее средство для манипуляций с данными древовидных структур. Например, чтобы перерисовать сложный многокомпонентный объект, достаточно подать его головному объекту-списку всего одну команду на перерисовку. Данная команда будет применена головным объектом к собственным данным и затем продублирована для подчиненных на всех уровнях вложенности.

2.4 Создание базисного элемента управления информационными процессами.

Как уже было сказано ранее, любой электронный объект, должен включать в себя две составных части: **данные** и программные **методы управления** этими данными.

С данными все ясно – это совокупность электронных объектов, составляющих тело электронной модели проекта. Составляющие тело проекта электронные объекты имеют четкую структуру и строго определенные собственные (внутренние) методы управления. В этом их сила, основа простоты исполнения и надежности в работе.

Со структурой общей электронной модели дело обстоит несколько сложнее. Общая модель обладает прямо противоположными свойствами: **структура ее не фиксирована и может меняться со временем**. Соответственно и методы управления такой структурой, и циркулирующие в ней информационные процессы не могут быть постоянны. И вообще: возможно ли в принципе создать электронную модель процесса? В отличие от данных, которые являются статическими **материальными объектами** (их можно редактировать и сохранять в файлах), информационный процесс (длящаяся во времени последовательность действий) выглядит субстанцией нематериальной. Можно ли создать материальную модель нематериального объекта? Оказывается – можно. Только сначала необходимо четко определиться с самим понятием информационного процесса: что это такое и каково его назначение?

Введем следующие определения:

- а) **Электронный информационный процесс есть функционирующий во времени набор программных методов.**
- б) **Назначение электронного информационного процесса – управление перемещением данных.**

Особо отметим, что из функций информационного процесса сознательно исключено понятие ‘генерация новых данных’, поскольку наш объект-процесс изначально позиционируется исключительно как управляющий элемент.

Такое определение существенно сокращает набор вменяемых процессу видов действий над данными, и главное, исключает из его функций необходимость анализа внутренней структуры данных. Построенную из таких объектов-процессов систему управления можно сравнить с почтовой службой: система опирается на строго определенную сеть адресатов, имеет синхронизированный во времени механизм получения, перемещения и выдачи корреспонденции. И при этом не вскрывает корреспонденцию, довольствуясь только указанными внешними параметрами.

В результате на долю информационного процесса остается вполне конкретный перечень действий, необходимых и достаточных для управления перемещением данных:

- **Получение исходных данных.** Действие, необходимое для начала любого процесса;

- **Сортировка данных.** Алгоритм работы напоминает алгоритм работы почтового сортировочного узла: по внешним атрибутам файла данных определяется тип данных (текст, модель, команды ...), степень готовности (входные, сгенерированные, согласованные, ...) и некоторые другие параметры;

- **Рассылка данных.** Исходя из этих параметров и общей логики работы процесса файлы данных отправляются по установленным для данного процесса адресам: на редактирование, согласование(утверждение) и далее на вход следующего процесса, либо обратно на доработку.

Можно утверждать, что подавляющее большинство из возможных манипуляций с данными укладывается в эту схему действий. А если это так, то не составляет большого труда создать базисный электронный объект, реализующий данный набор действий. Этот объект будет инвариантен по отношению к виду данных, поскольку его главная задача – рассылка данных. Генерация новых данных осуществляется в многочисленных графических меню-интерфейсах, и не являются функцией управляющих процессов.

Отметим, что базисный электронный объект управления (процесс) является производным от базисного элемента представления данных ЕИП и наследует все его свойства. Поэтому и работать с ним можно как с обычным графическим электронным объектом. На его основе можно строить **процессные сборки – последовательности связанных информационных процессов**, лежащие в основу все более сложных и разветвленных процессов вышележащих уровней.

3. Оценка трудоемкости задачи

3.1 Оценка трудоемкости работ по созданию программного ядра ЕИП.

Озвученная в самом начале идея о создании электронного макро-объекта (каковым является ЕИП), заменяющего собой множество существующих специализированных программных комплексов, на первый взгляд может показаться абсурдной и невыполнимой. Ведь если принципиальную возможность создания единой формы представления данных, в свете вышеизложенного, уже трудно оспорить, то остается другой вопрос – а сколько времени и сил потребуется на реализацию данной идеи? Действительно, над созданием существующего ПО не один год трудились большие коллективы разработчиков. Были задействованы серьезные интеллектуальные ресурсы для сбора и систематизации исходных данных, для построения соответствующих математических моделей. Где взять силы и средства для выполнения таких объемов работ?

Попробуем разобраться, насколько затратен на самом деле процесс создания программного узла ЕИП?

- 1) **Интеллектуальные затраты.** Все алгоритмические и функциональные описания математических моделей уже разработаны и обкатаны в процессе создания существующего ПО. В модели ЕИП могут быть использованы с минимальной коррекцией.
- 2) **Математическая модель.** При внимательном рассмотрении оказывается, что все великое множество существующих на текущий момент математических моделей (если абстрагироваться от типов хранимых ими данных), основано всего на двух видах представления данных: списочном и табличном. Причем табличное представление данных является всего лишь менее эргономичной модификацией списочного представления. Поэтому используемое в модели ЕИП списочное представление данных позволяет без особого труда реализовать функционал существующего ПО.
- 3) **Меню-интерфейс.** Поскольку в ЕИП существует единая форма представления данных, то и механизм визуализации этой формы – тоже один. Причем он достаточно прост в исполнении, т.к. реализует ограниченный набор простых манипуляций с обезличенными графическими объектами.
- 4) **Компоненты математической модели.** Действительно, число конкретных видов электронных объектов, используемых в существующем ПО – огромно! Но при

проектировании ядра ЕИП не ставится задача создания моделей всех возможных видов объектов. **Задача** ставится по-другому – **создать один, но универсальный электронный компонент!** Свойства такого компонента представлены в п. 2.3.1.

- 5) **Компоненты системы управления данными.** Описанный в п. 2.4 элемент управления (процесс) является базисным компонентом единой распределенной системы управления информационными процессами в рамках ЕИП. Заложенные в нем программные методы управления данными позволяют реализовывать все функции существующих MES/ERP/PDM/PLM и других систем управления (тем более что функционально все эти системы – близнецы-братья).
- 6) **Хранилища данных.** В системе ЕИП хранилище данных вообще не является самостоятельным электронным объектом. Это – обычный программный узел ЕИП, в котором отключены часть проектирующих программных методов, и наоборот, задействованы дополнительные функции по классификации, сортировке и размещению данных. В этом смысле хранилище данных – одна из электронных моделей в составе ЕИП.

Из вышеизложенного следует вывод: **создание программного узла ЕИП – обычная программистская задача средней сложности!** Главная трудность заключается не в программировании, а в осмыслении того, как должна выглядеть универсальная и не противоречивая форма представления данных. Практическая же реализация оказывается тем проще, чем глубже осмысление структуры создаваемого продукта.

3.2 Оценка трудоемкости создания электронных моделей объектов и процессов.

При оценке трудоемкости следует вспомнить, что новый электронный компонент практически всегда является ‘наследником’ какого-либо из уже существующих объектов, приобретая весь багаж уже проверенных, функционально законченных программных методов, полностью обеспечивающих функционирование нового объекта в рамках программного узла ЕИП.

В результате оказывается, что весь объем работ по созданию нового электронного объекта сводится к коррекции существующих наследуемых параметров (графики, формы, привязки ...) и к реализации ограниченного набора дополнительных свойств и методов, коими собственно и отличается новый объект от всех остальных. Причем, по мере наполнения общей библиотеки электронных объектов трудоемкость их создания снижается, т.к. если разнообразие внешних форм объектов практически неисчерпаемо, то набор их функциональных возможностей – конечен.

Формирование библиотек электронных объектов ЕИП должно идти путем, аналогичным процессу создания открытого программного обеспечения - участвовать могут все пользователи ЕИП. Для этого в ядре ЕИП реализован развитый программный инструмент, позволяющий создавать самые причудливые электронные модели. Так же, как и при создании открытого ПО, формирование ЕИП может концентрироваться вокруг ‘центров кристаллизации’ – организаций, сопровождающих наиболее крупные и распространенные электронные модели (электронное предприятие, электронное правительство и т.п.). Но, в отличие от технологии создания открытого ПО, при формировании ЕИП максимально ограничивается элемент анархии при создании электронных компонентов моделей. Достигается это путем использования единого программного инструментария (с закрытыми текстами базовых элементов) и едиными правилами организации движения потоков данных в рамках ЕИП.

4. Типовая модель “Электронное предприятие”

Маленький сюрприз для скептиков. Основной элемент единого информационного пространства (программный узел ЕИП) уже реально существует! Многочисленные утверждения типа: “это невозможно”, “а это слишком сложно”, можно легко проверить. Просто запускаем это самое “невозможно” и пробуем в работе то, которое “слишком сложно”.

В представленных ниже приложениях показаны экранные снимки двух основных рабочих окон программного узла ЕИП.

Главное окно (Приложение 1), условно можно определить как окно управления статическими данными проекта, своего рода информационную систему. Через это окно осуществляется управление общей архитектурой моделируемого объекта. Это инструмент руководителя, позволяющий отслеживать текущее состояние, как всего объекта, так и отдельных его частей.

Процессное окно (Приложение 2), управляет динамической частью проекта. В нем сосредоточены функции по созданию и управлению всеми информационными процессами моделируемого объекта. Это инструмент разработчика. С его помощью формируется логика будущих информационных процессов, разрабатываются и распределяются исполнителям задания, назначаются виды работ и сроки исполнения. Здесь же осуществляется контроль работающих информационных процессов.

5.1. Визуализация структуры моделируемого объекта.

Главное окно, включает в себя два подокна, представляющих два вида отображения общей структуры модели.

В левом подокне структура моделируемого объекта представлена в виде дерева подчинения объектов модели (дерева иерархий). Перемещаясь по данному дереву, можно не только путешествовать по административной структуре предприятия, но и получать доступ к внутренним данным узлов ЕИП с любой разрешенной степенью детализации.

Правое подокно представляет структуру предприятия в ‘матричном’ виде. Матричное представление не такое детализированное, зато оно позволяет визуализировать в одном окне сразу все основные элементы дерева иерархий, и получать доступ к любому объекту ‘за один клик’.

Особо следует отметить: понятие ‘получить доступ’ вовсе не означает, что кто-либо, находясь в одном узле ЕИП, может физически проникнуть внутрь другого узла. Такое в ЕИП в принципе невозможно! Дело в том, что **отображенная в данном окне структура – всего лишь макет** реального размещения данных в ЕИП. На самом деле все **элементы главного окна – это отображения** реальных узлов ЕИП (отдельных ПЭВМ), разнесенных территориально по разным физическим адресам. Доступ же к информации данных узлов осуществляется путем внутрисетевых обращений по схеме ‘запрос-ответ’ через общедоступный почтовый ресурс ЕИП. Данная схема взаимодействия, с одной стороны, полностью исключает случайный несанкционированный доступ, а с другой – позволяет в любой момент по запросу получить любые (разрешенные) данные с любого узла ЕИП. Под запросом в данном случае понимается простой ‘клик’ мышью по выбранной картинке, либо по элементу окна дерева иерархий.

Механизм макетирования является важнейшим инструментом управления распределенной электронной моделью. Поскольку элементы макета и их реальные прототипы в узлах ЕИП находятся в постоянном взаимодействии, любые манипуляции с элементами на макете отображаются в реальных узлах ЕИП, и наоборот.

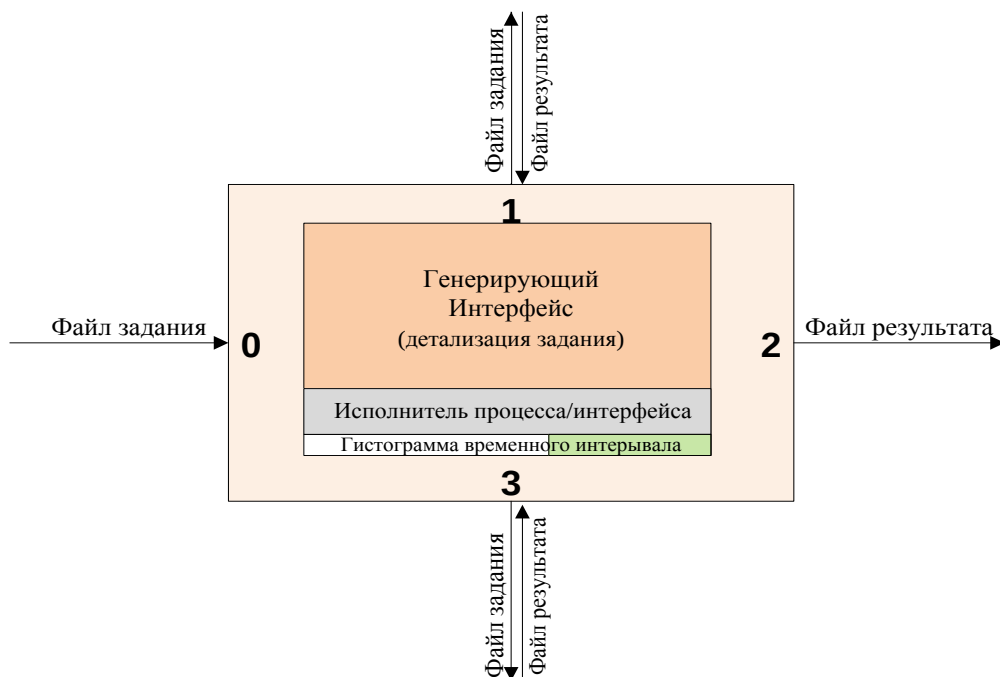
5.2. Визуализация управляющих информационных процессов.

Как уже было сказано ранее, **жизненный цикл** любой электронной модели включает два вида деятельности: **создание новых данных и управление движением данных**. Если для создания и редактирования данных вполне достаточно средств управления деревом иерархий, то **управление движением данных** требует специального объекта управления и специального пользовательского интерфейса (Приложение 2).

5.2.1 Базисный информационный процесс

В основе построения сложных информационных процессов (процессных сборок) лежит специальный базисный электронный объект – процесс. Физически это стандартный базисный объект хранения данных, дополненный программными методами управления перемещением данных.

Процесс управления заданиями



Кроме того, данный объект способен функционировать в режиме реального времени и оснащен механизмом программируемой рассылки данных внутри структуры ЕИП.

5.2.2 Процессная сборка

Под этим термином понимается совокупность единичных процессов, связанных в единую логическую схему для реализации поставленной задачи.

Так же, как и базисный объект хранения, объект-процесс 'в чистом виде' представляет собой абстрактный объект (безликий строительный 'кирпичик' процессной сборки). Тем не менее, из таких 'кирпичиков' уже можно строить сложные блок-схемы, устанавливая пока еще абстрактные логические связки между их элементами. В этом заключается вся прелесть использования абстрактных объектов: возможность строить сложные логические конструкции информационных процессов, не вдаваясь в содержание данных, которыми им предстоит манипулировать. На практике оказывается, что различных видов абстрактных логических схем перемещения данных не так уж и много. Многообразие рождается, когда мы начинаем наполнять абстрактные логические схемы конкретным содержанием.

В нашем случае абстрактный объект-процесс превращается в конкретный управляющий объект, когда к нему 'прикрепляется' **генерирующий** конкретные данные **меню-интерфейс**. Этими данными наш процесс теперь будет управлять, и по бывшим абстрактным связкам начнут передаваться уже конкретные данные. В результате (после конкретизации всех элементов) абстрактная логическая конструкция приобретает свое уникальное 'лицо'. В приложении 2 представлена блок-схема сложной процессной сборки (сквозной цикл проектирования электронного узла), где в малых окнах показаны в развернутом виде некоторые прикрепленные меню-интерфейсы.

Если в данной блок-схеме заменить некоторые прикрепляемые интерфейсы другими, она легко может превратиться, например, в процесс проектирования узла машиностроения, или в архитектурный проект.

5.2.3 Запуск информационных процессов

Даже после обретения процессной сборкой конкретного содержания, она еще не является полноценным информационным процессом. Во-первых, потому, что для ее реализации еще не назначены исполнители. И, во-вторых, потому что данная конструкция пока что является только макетом реального процесса, выполненном на макете электронной модели предприятия.

Для того чтобы макет превратился в реальность, остается произвести два простых действия:

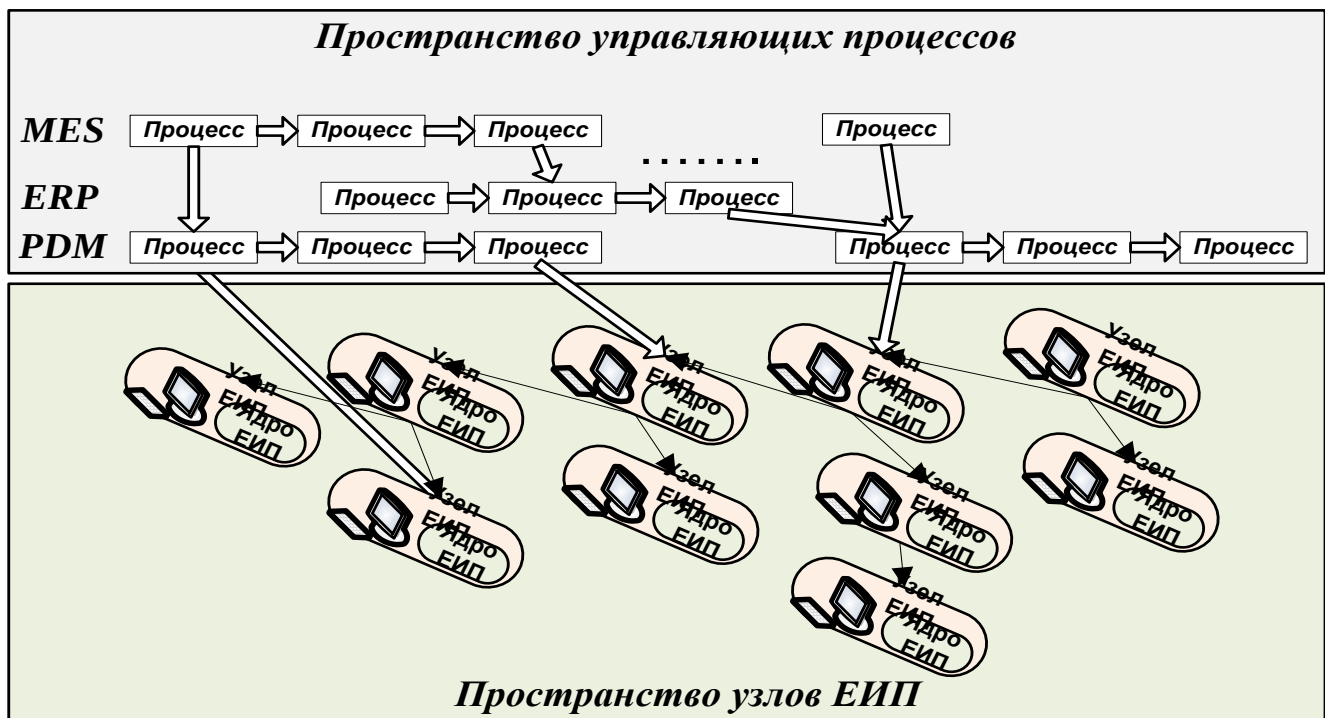
- Назначить всем элементам процессной сборки исполнителей (простым перетаскиванием имени исполнителя из древовидного списка левого окна в соответствующий процесс);
- Активировать кнопку “PLACE” рабочего меню-интерфейса (копии элементов сборки будут разосланы на ПЭВМ узлов-исполнителей);
- Активировать кнопку “START” рабочего меню-интерфейса (исполнителям будет разослана команда активации попавших в их зону ответственности участков процесса).

После этого запущенные элементы процессной сборки (процессы) в каждом узле ЕИП переходят в режим автономного функционирования. Они автоматически активируются при включении ПЭВМ, получают, обрабатывают, отправляют данные, деактивируются при выключении ПЭВМ или по завершении решаемой задачи.

5.2.4 Совместное функционирование информационных процессов

Ранее было отмечено, что архитектура ЕИП имеет процессно-ориентированный характер. Теперь можно расшифровать это понятие. Термин ‘процессно-ориентированный’ означает, что движением информационных потоков в общей модели управляют не объекты административной структуры (руководя информационными процессами в рамках своей подчиненной административной структуры), а наоборот, информационные процессы, (функционирующие на пространстве всей модели и управляющие административными объектами в рамках выполняемого процесса).

Схематично процессно-ориентированную модель управления информационными процессами можно представить в виде многослойной объемной структуры.



Нижний слой составляет распределенная система хранения данных (узлы ЕИП). В плоскости этого слоя организована иерархическая структура административного деления предприятия (дерево иерархий). Следующие слои образованы множеством информационных процессов различных уровней (PDM/PLM – управление проектированием, MES/ERP- управление ресурсами, финансами и т.д.). Множество процессов существует как система независимых образований, находящихся в одном пространстве, но не взаимодействующих между собой и с узлами ЕИП до тех пор, пока между ними в нужных местах не будут организованы горизонтальные и вертикальные связи.

ЗАКЛЮЧЕНИЕ

Если рассматривать динамику любого процесса развития с философской точки зрения, можно сформулировать следующее определение: **процесс развития есть стремление к идеалу**. Причем на определенных этапах (витках спирали развития) этот идеал – достижим. Например, в процессе достижения наиболее эргономичной формы физического тела (какими бы путями мы ни шли), мы неизбежно придем к единственной (для трехмерного пространства) идеальной форме – форме шара.

В мире электронных моделей тоже существует понятие идеала, к достижению которого (порой неосознанно) стремятся все, кто работает в области информационных технологий. Развитие и конкуренция различных операционных систем, платформ и языков программирования, все направлено на достижение одной цели – созданию идеальной электронной формы представления данных. Причем, при сравнении конкурирующих видов программного обеспечения, можно отметить любопытную особенность – конкуренты становятся все более похожими друг на друга. И из совокупности их общностей постепенно начинают вырисовываться контуры того идеального построения информационной среды, которая полностью удовлетворила бы потребности современного общества.

Применительно к обсуждаемой теме, можно утверждать, что предлагаемые принципы построения электронной модели единого информационного пространства, если и не идеальны, то близки к идеалу. Основанием для такого утверждения может служить тот факт, что при проектировании ЕИП основные архитектурные решения позаимствованы у самой Природы, а методы формирования и взаимодействия данных основаны на проверенных жизнью физических и математических закономерностях.

Представленный вариант реализации ЕИП тоже еще очень далек от совершенства. Потребуется приложить еще немало сил для того, чтобы придать ему законченный вид, наработать необходимое количество электронных компонентов, разработать и отладить недостающие программные методы, произвести другие необходимые работы. Но это всего лишь техническая, рутинная сторона вопроса.

Главное состоит в другом: само наличие варианта физической реализации электронной модели ЕИП не оставляет больше ни сомнений, ни ‘белых пятен’ в вопросе о принципиальной возможности создания универсальной модели представления данных.

Представленный вариант реализации программного узла ЕИП в настоящий момент можно рассматривать как первый шаг на пути создания полноценной модели ЕИП. Многое еще предстоит сделать, но результат стоит того: **владелец программного продукта такого уровня обеспечит себе доминирующее положение на рынке ПО на многие годы вперед!**

Подробнее о проекте см. на сайте **sprspase.16mb.com**

