

УДК 004.822

В.А. Кочуров

МЕТОДИКА СОЗДАНИЯ БАЗ РАСПРЕДЕЛЕННЫХ ИНЖЕНЕРНЫХ ЗНАНИЙ

Рассматриваются принцип и приемы создания моделей распределенных инженерных знаний, а также программная реализация таких моделей в форме сервера информационно-логических таблиц (СИЛТ). Благодаря взаимодействию СИЛТ между собой на основе технологии WCF (Windows Communication Foundation) создается кластер СИЛТ, посредством которого реализуется локальное пространство знаний компании. Кроме того, в связи с наличием специального веб-приложения, являющегося частью данного комплекса, создается возможность доступа к кластеру СИЛТ с любого устройства, имеющего веб-браузер и доступ в сеть Интернет.

Введение

В основе создания новых образцов промышленной продукции в современных условиях лежит понятие жизненного цикла изделия как совокупности этапов, через которые проходит изделие за время своего существования: маркетинговые исследования, составление технического задания, проектирование, технологическая подготовка производства, изготовление, поставка, эксплуатация, утилизация.

Каждый из составляющих жизненный цикл изделия этапов представляет собой бизнес-процесс, содержащий две составляющие:

- интеллектуальную деятельность, в процессе которой постепенно формируется образ желаемого результата, оформляемый затем в форме комплекта документации;
- физическую деятельность по реализации образа результата на основе созданного комплекта документации.

В основе интеллектуальной деятельности в бизнес-процессах явно или неявно лежит проектирование сценария последующей физической деятельности. Другими словами, основу любой производственной деятельности составляют процессы проектирования не только конечного результата этой деятельности (например, выпускаемого продукта), но и возможных сценариев выполнения этой деятельности (совокупности бизнес-процессов, обеспечивающих выпуск конечного продукта).

Независимо от сложности или объемности какого-либо процесса проектирования он распадается в конечном итоге на множество элементарных шагов – состояний процесса. Поскольку каждое состояние должно тем или иным способом фиксироваться в документе определенной формы, то элементы информации, характеризующие образы создаваемых конструкций или процессов в каждом состоянии их разработки, мысленно объединяются проектировщиком или аналитиком в информационные модели состояний процесса проектирования. Переход от одного состояния к последующему состоянию осуществляется посредством принятия решений. Под решением здесь понимается либо конкретный элемент создаваемой информации, либо указание, какой элемент информации необходимо создавать в данных условиях и как его создавать.

Таким образом, можно сделать важный вывод о том, что как в процессе проектирования конечного продукта, так и при реинжиниринге бизнес-процессов создания этого продукта исполнители принимают решения, используя определенную информационную среду. Отсюда возникает вопрос: что такое информационная среда принятия решений и как ее моделировать в программной реализации систем поддержки принятия решений?

1. Базы знаний в процессах проектирования

Процесс проектирования (или реинжиниринга) можно рассматривать как комплекс взаимосвязанных работ по созданию, дополнению и преобразованию образа объекта проектирования (изделия или процесса) со всей необходимой для целей его изготовления (реализации) или обслуживания информационной полнотой. Информацию, определяющую образ объекта на лю-

бой стадии проектирования и зафиксированную в любой из общеупотребительных форм (графической, технической, математической, устной и т. д.), можно назвать информационной моделью состояния процесса проектирования.

Рассмотрим информационную среду процесса проектирования на примере технологического проектирования. Этот процесс протекает в информационной среде, содержащей исходную и производную информацию. В свою очередь, исходная информация делится на переменную и условно постоянную информацию, а последнюю можно разделить на справочную и методическую информацию.

Под исходной информацией понимается информация, существующая до начала проектирования. Переменная часть этой информации содержит необходимые сведения об изделии, программе выпуска и т. п. и подготавливается для каждого конкретного процесса проектирования.

Условно-постоянная справочная информация изменяется эпизодически независимо от конкретного процесса проектирования и содержит сведения о методах получения заготовок, обработки и т. п.

Производная информация формируется в процессе технологического проектирования. Она содержит необходимые сведения о маршруте обработки отдельных поверхностей, промежуточных конфигурациях детали в процессе обработки и т. п.

Методическая информация определяет причинно-следственные связи между элементами исходной и производной информации, а также содержит рекомендуемые решения, которые следует принимать при определенных условиях на определенных стадиях проектирования.

В ходе выполнения любой работы, составляющей процесс проектирования, происходит переход от предшествующего к последующему состоянию процесса посредством использования методической информации, которая определяет способ преобразования текущего состояния в новое состояние и обычно не фиксируется в документах (остается «в тени»).

Учитывая вышесказанное, в настоящем контексте можно ограничить трактовку термина «знание» и рассматривать его как рекомендацию выполнить определенное действие в определенных условиях, т. е. знание представляет собой сопоставленную состоянию рекомендацию способа перехода к последующему состоянию. Такое ограничение трактовки термина «знание» позволяет уточнить понятие «база знаний» и рассматривать ее как базу данных, логическая модель которой представляет собой множество состояний какого-либо процесса совместно с правилами перехода к последующему состоянию этого процесса. Другими словами, рассматриваемые нами знания – это эвристические правила преобразования текущего состояния модели в новое состояние.

Можно видеть, что упомянутая выше методическая информация представляет собой концептуальную базу знаний над базой фактов, причем в качестве базы фактов здесь выступает исходная, условно-постоянная и производная информация [1].

Теперь важно определить способ формализованного представления инженерных знаний [2]. Традиционно такие знания представляются посредством описаний, которые могут содержать текст, формулы, таблицы, рисунки (эскизы), чертежи, трехмерные модели, скрипты, программные коды. Данный способ представления знаний можно назвать наиболее предпочтительным, поскольку он обладает главными преимуществами – доступностью и понятностью конечному пользователю-инженеру (лицу, принимающему решения).

Посредством современных систем программирования [3] можно запрограммировать любой сценарий, однако знания, представленные в форме программного кода, могут использоваться только внутри самой программы, и поэтому такой способ следует считать неприемлемым. Модель представления инженерных знаний должна обеспечивать возможность взаимодействия с ней инженеру в традиционной манере, программисту, разрабатывающему программные решения, а также программе, имитирующей интеллектуальную деятельность.

2. Информационно-логические таблицы

В настоящей работе рассматривается опыт использования в качестве моделей представления инженерных знаний информационно-справочных таблиц (рис. 1), которые широко применяются в учебниках, справочниках, а также различной справочно-нормативной документации.

Любая информационно-справочная таблица является структурно организованным набором совместно используемых фактов. Каждый факт характеризуется именем, значением и отношением к соседним фактам, причем это отношение выражается взаимным расположением данного факта относительно остальных фактов набора.

Именно расположение факта относительно других фактов набора является семантически нагруженным, позволяет выделить смысл каждого факта – его интерпретацию. Структурно организованный набор фактов информационно-справочной таблицы представляет собой определенную сущность – абстрактный объект.

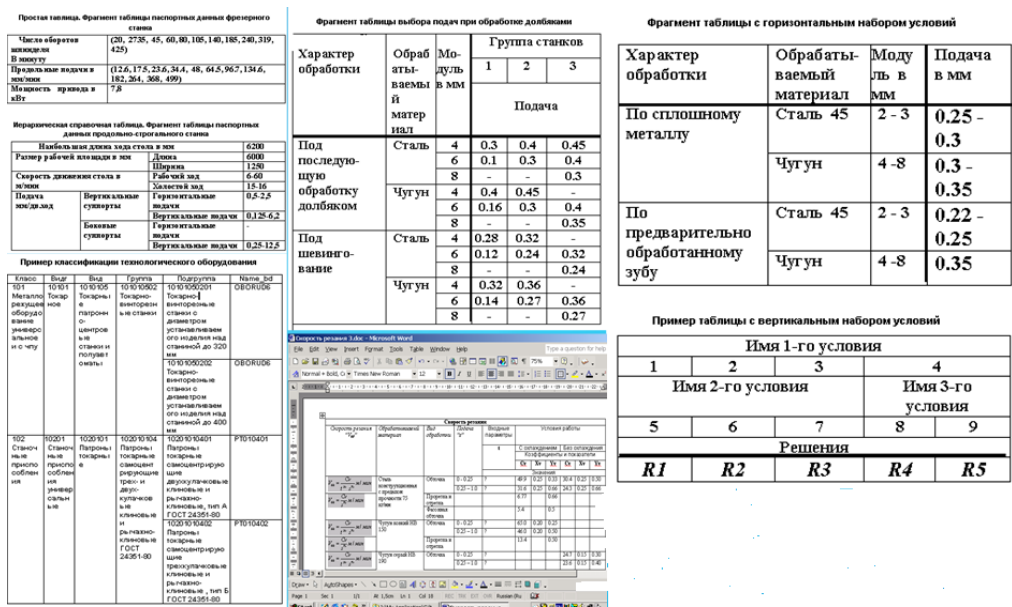


Рис. 1. Примеры информационно-справочных таблиц

Поскольку взаимное расположение фактов набора и соответствующую семантику может распознавать только человек, использование информационно-справочных таблиц для представления эвристических знаний в программных реализациях основывается на преобразовании их в так называемые информационно-логические таблицы (ИЛТ) посредством введения структурно-семантических ролей, которые сохраняются в контексте формализованного представления модели.

Структурно-семантические роли (рис. 2) имеют особое значение в использовании ИЛТ. Основная часть списка ролей содержит информацию о структурной организации ИЛТ и формируется автоматически программным способом при занесении таблицы в базу данных. Затем новые роли могут добавляться оператором, привнося в ИЛТ ту или иную дополнительную семантику.

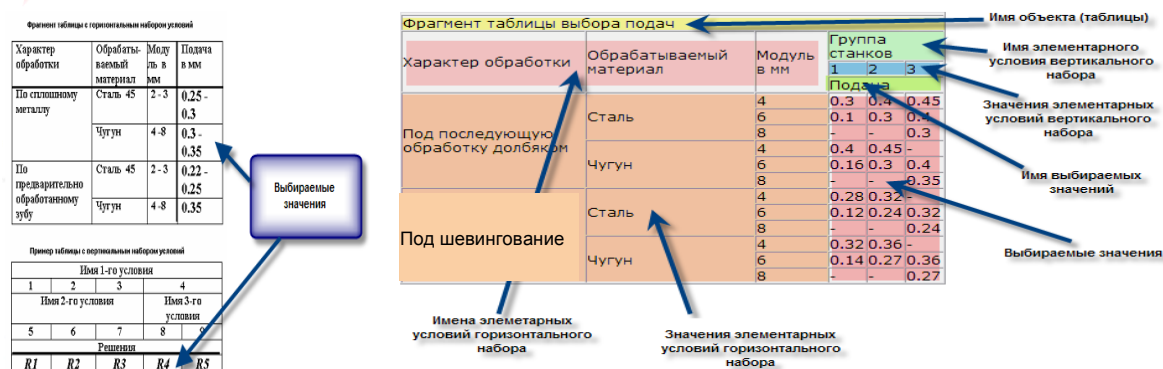


Рис. 2. Пример структурно-семантических ролей, преобразующих информационно-справочные таблицы в информационно-логические

После проведения формализации модель любой ИЛТ независимо от ее внешнего вида может рассматриваться на формальном уровне как объект, состоящий из упорядоченной коллекции именованных фактов-контейнеров, каждый из которых содержит коллекцию фактов. При этом каждый факт также представляет собой объект, определяемый фиксированным набором свойств: идентификатор, имя, значение, список ролей, результат выполнения операции, ссылка на визуальный элемент управления, ссылка на родительский контейнер, ссылка на контейнер верхнего уровня.

Если ИЛТ содержит атрибут с ролью «выбираемое значение», то она может интерпретироваться как однородная группа правил «если – то». Определение «однородная» означает, что выбираемые решения являются элементами объема одного и того же понятия: диаметра, длины, допустимого значения инвестиций и т. п. В этом случае следует обратить внимание на то, что атрибуты в условии, определяющем выбор значения (заклЮчения правила), являются атрибутами одной или нескольких независимых сущностей информационной среды принятия решений. Следовательно, ИЛТ, содержащая атрибут с ролью «выбираемое значение», выражает скрытое причинно-следственное отношение между значениями характеристик одной или нескольких независимых сущностей определенной предметной области. Именно это замечание является важным аргументом в пользу использования ИЛТ для моделирования информационной среды бизнес-процессов.

3. Реляционно-предикативная иерархическая модель базы данных

Информационную среду некоторого бизнес-процесса будем рассматривать как определенную предметную область. Концептуальную модель такой предметной области составляет множество сущностей (абстрактных объектов), находящихся между собой в различных отношениях (не обязательно теоретико-множественных).

Каждая сущность отличается от других сущностей составом своих постоянных свойств (характеристик). В свою очередь, каждое свойство объекта (сущности) представляется атрибутом, который характеризуется именем и набором (множеством) допустимых значений, а также атрибутивным предикатором, который определяет отношение между именем и значением (например, «это есть то»). Состав свойств сущности часто называют типом.

Учитывая, что каждый тип представлен именем и составом атрибутов, каждый атрибут также представлен именем атрибута, предикатором и множеством допустимых значений и, наконец, каждое значение представлено автономно определенной лексемой, можно сделать вывод, что базовый, или нижний, уровень модели предметной области составляет множество лексем трех категорий:

- имена;
- значения;
- предикаторы.

В свою очередь, лексемы объединяются в структуры (типы) – поименованные наборы лексем.

Казалось бы, отсюда следует, что на нижнем уровне иерархии модели некоторой предметной области необходимо создать представление трех множеств лексем: имен, значений и предикаторов. Однако этот вывод верен лишь частично. Очень часто лексемы в одном контексте выступают в роли значения, а в другом – в роли имени. Например, в предложении «Вид поверхности есть цилиндр» слово «цилиндр» выступает в роли значения. Эта же лексема в предложении «Цилиндр есть поверхность, образованная...» выступает в роли имени. Поэтому лексемы, представляющие имена атрибутов, элементов и объектов, а также значения атрибутов, целесообразно занести в общую таблицу лексем (тезаурус, словарь).

Примечание. Отличие множества значений от множества имен состоит лишь в том, что значениями могут быть типы больших значений (например, большие тексты) или двоичные файлы (изображения и т. п.). Однако большая часть значений – это короткие лексемы: числа, строки символов, даты. Отсюда следует, что таблицу имен можно объединить с таблицей значений, добавив в таблицу предикаторов роль «имя атрибута».

Логическая модель базы данных для хранения определенных выше моделей знаний в форме ИЛТ основывается на понятии «факт». Факт – это истинное по определению высказывание, утверждающее, что представленное определенной лексемой значение является значением атрибута, имя которого определяется второй лексемой. При этом между данными лексемами существует отношение, определяемое атрибутивным предикатором. Образованный таким способом атрибут представляет значение свойства элемента в составе определенного объекта указанного типа.

Факты объединяются в контейнеры нижнего уровня – сложные атрибуты и элементы. Контейнеры, представляющие элементы, объединяются в контейнеры – объекты. Контейнеры, представляющие сложные атрибуты, элементы и объекты, будем называть базовыми контейнерами. Помимо базовых контейнеров могут существовать производные контейнеры на уровнях между элементами и объектом, а также на уровне объектов.

Логическая модель реляционной базы данных (рис. 3) содержит:

- таблицу значений (SiltValues), представляющую тезаурус;
- таблицу предикаторов (SiltPred), содержащую структурно-семантические роли;
- таблицу фактов (SiltFkt), представляющую структуру объекта (ИЛТ), поскольку помимо отдельных фактов она содержит и сведения о структурных связях между фактами. Эта таблица определяет факты вне контекста, в котором они встречаются в исходном документе. Однако некоторый атрибут, содержащийся, например, в информационно-логической таблице, имеет дополнительные свойства, такие как номер столбца, номер строки источника, кратность по строке и столбцу. Необходимость в этих сведениях возникает при программной визуализации ИЛТ. Поэтому вводится еще одна таблица – SiltStruct, содержащая сведения о свойствах факта в контексте первоисточника.

Один и тот же факт может играть несколько ролей в одно и то же время в семантической модели документа-источника, что вынуждает вводить еще одну таблицу – SiltRole, содержащую роли, которые играет факт в контексте первоисточника. Эта таблица представляет отношение «один ко многим» между таблицами фактов и предикаторов (SiltFkt и SiltPred).

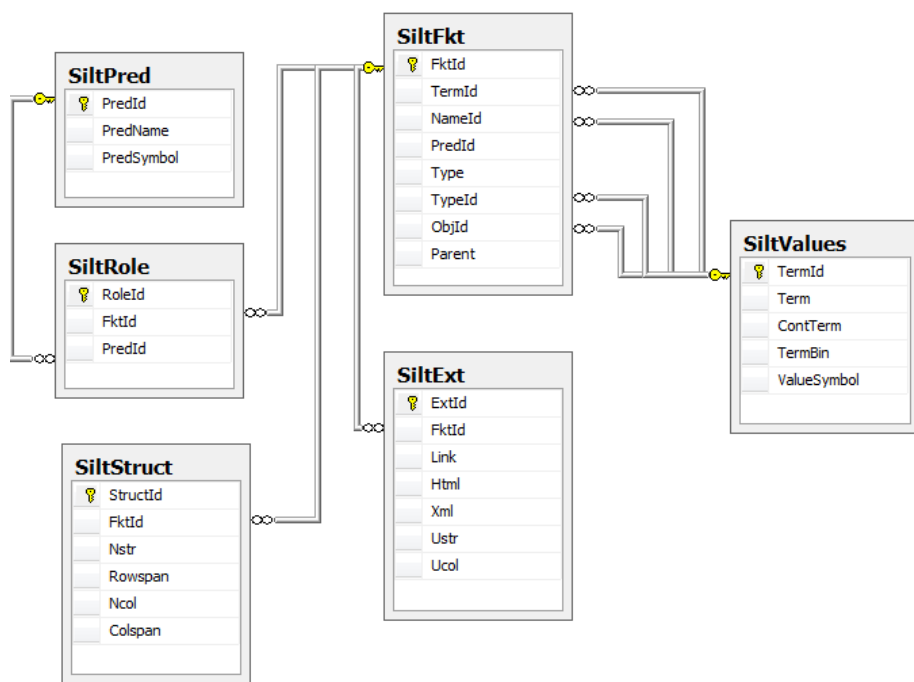


Рис. 3. Логическая схема базы данных для реляционно-иерархической предикативной модели структурных объектов

Остаются неучтенными еще несколько дополнительных сведений, связанных с фактом и его представлением:

- лексема может представлять ссылку на другой факт;
- лексема может потребовать HTML-форматирования при ее представлении в браузере;
- лексема может являться именем объекта, который требует представления в виде полноценной веб-страницы;
- если лексема представляет имя объекта, то для него может потребоваться XML-представление.

Отсюда следует необходимость введения еще одной таблицы – SiltExt, обеспечивающей хранение перечисленных выше сведений.

Схема полученной таким образом базы данных представлена на рис. 3.

Необходимо указать на главное отличие предлагаемого подхода от традиционной реляционной модели. В классической нормализованной реляционной модели каждая сущность (в нашем случае ИЛТ) представляется реляционной таблицей. В предлагаемом подходе реляционные таблицы используются для представления словарей лексем, предикаторов и фактов, что позволяет построить универсальную физическую модель базы данных, в которую можно «вкладывать» различные логические модели предметных областей.

Замечание. ИЛТ как самостоятельная сущность в рассматриваемой логической модели базы данных отсутствует, поэтому для взаимодействия с такой базой данных требуется специальный программный интерфейс.

4. Сервер информационно-логических таблиц

В настоящей работе рассматривается один из вариантов программной реализации взаимодействия с базой данных, описанной в разд. 3, посредством программно-информационного комплекса СИЛТ. СИЛТ – активный сервер приложений, управляющий базами информационных ресурсов (БИР), объединяемых по функционально-структурным признакам. Например, это могут быть информационные ресурсы для решения задач статистического анализа, обеспечения расчета по деталям машин или решения задач оптимизации в определенной предметной области.

БИР – это структурно организованная коллекция объектов предметной области, которые могут представлять модели знаний в форме ИЛТ, простые текстовые документы, документы WORD/EXEL, рисунки различных форматов, 2D- и 3D-модели САD-систем, исходные тексты и исполнимые файлы программ, сценарии в форме скриптов. Следует заметить, что интерфейс любого такого объекта обеспечивает ИЛТ, посредством которой устанавливаются логические связи как между самими объектами, так и между отдельными элементами этих объектов.

БИР размещается в контейнере, в качестве которого выступает реляционная база данных под управлением любой коммерческой СУБД. Логические модели всех таких баз данных идентичны и представляют собой рассмотренную выше реляционно-иерархическую предикативную модель структурных объектов. Эти два фактора определяют преимущества предлагаемой методики.

5. Функциональная модель СИЛТ

В структурном плане СИЛТ содержит центральное ядро, представленное обработчиком информационно-логических таблиц, который реализован в форме WCF-службы (Windows Communication Foundation Services). Данная служба экспонирует объекты, находящиеся под управлением СИЛТ. Кроме того, СИЛТ содержит WCF-клиента, который обеспечивает доступ к информационным ресурсам других серверов (рис. 4).

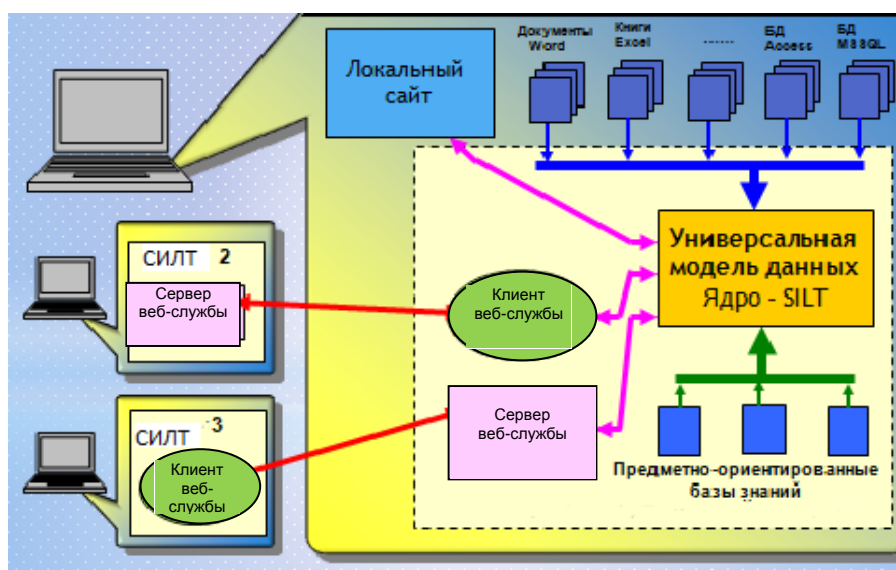


Рис. 4. Архитектура сервера информационно-логических таблиц

Из изложенного выше следует, что структура функциональной модели СИЛТ содержит три категории функциональностей:

- функциональное ядро;
- визуальный интерфейс;
- интерфейсы удаленного доступа.

Функциональное ядро инкапсулирует всю целевую функциональность СИЛТ на уровне программного интерфейса:

а) функциональность взаимодействия с БИР:

- создание новой БИР;
- открытие существующей БИР;
- просмотр состава ИЛТ в открытой БИР;
- выбор и открытие конкретной ИЛТ;
- сохранение ИЛТ в заданной БИР;

б) манипулирование ИЛТ:

- создание новой ИЛТ и сохранение ее в заданной БИР;
- удаление из БИР существующей ИЛТ;
- произвольное редактирование структуры ИЛТ;
- редактирование контекста факта (чтение и изменение ролей, принадлежности родительскому элементу; определение входных и выходных параметров);
- реализацию вычислений посредством встроенного скриптового языка и встроенного программного интерфейса к продукту MathCad.

Визуальный интерфейс СИЛТ представляет доступ со стороны конечного пользователя к функциональности ядра как локального, так и удаленного СИЛТ-сервера; интерфейсы удаленного доступа обеспечивают удаленный доступ СИЛТ-клиента к СИЛТ-серверу, внешним SQL-базам данных, Microsoft Office Word и MathCad.

Процесс доступа к удаленному СИЛТ-серверу реализуется посредством следующего сценария:

- на компьютере-хосте запускается СИЛТ-сервер;
- на компьютере-клиенте открывается СИЛТ-клиент в локальном режиме;
- посредством визуального интерфейса СИЛТ-клиента выбирается открытый на хосте СИЛТ-сервер из списка доступных СИЛТ-серверов;

– в результате выбора на СИЛТ-клиенте нужной БИР СИЛТ-сервера автоматически посылается запрос на СИЛТ-сервер, который возвращает СИЛТ-клиенту список ИЛТ выбранной БИР;

– далее все действия аналогичны работе в локальном режиме за исключением того, что команды будут выполняться на СИЛТ-сервере.

Благодаря реализации взаимодействия СИЛТ-серверов между собой создается сеть СИЛТ (кластер СИЛТ). Внутри такого кластера каждый СИЛТ может предоставлять свою функциональность другим серверам и одновременно является клиентом, обращающимся к функциональности других СИЛТ. Таким образом, создается локальное пространство знаний компании (рис. 5).

Помимо взаимодействия СИЛТ-серверов между собой внутри кластера посредством специального веб-приложения, являющегося частью данного комплекса, создается глобальная структура взаимодействия, которая позволяет получить доступ к кластеру СИЛТ из любой точки земного шара, с любого устройства, имеющего веб-браузер и доступ в сеть Интернет (рис. 6).

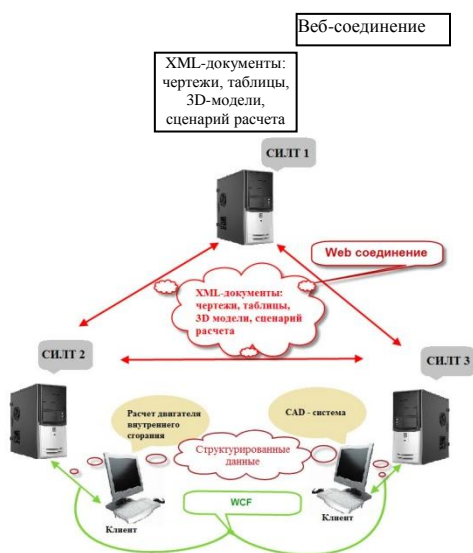


Рис. 5. Кластер СИЛТ-серверов

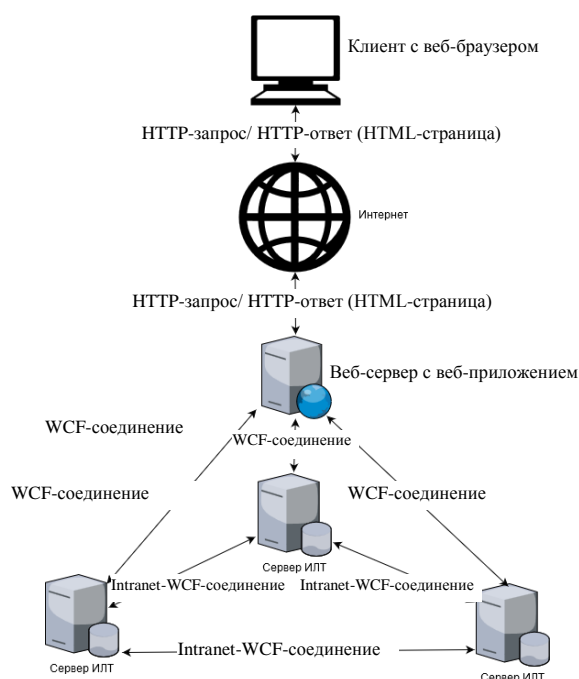


Рис. 6. Схема доступа к кластеру СИЛТ-серверов посредством веб-обозревателя

Целью интерфейса удаленного доступа к внешним базам данных является привязка аргументов вычислительных ресурсов СИЛТ непосредственно к данным внешнего источника, представленного SQL-базой данных. Эта функциональность реализуется под управлением эксперта (администратора) без участия программиста. Создание запросов к базе данных выполняется с помощью конструктора запросов с дружественным интерфейсом, который помогает пользователю самостоятельно получить необходимые ему данные из SQL-базы данных.

Целью интерфейса удаленного доступа к Microsoft Word является автоматизация процесса переноса информации из документов Microsoft Word в базы ресурсов СИЛТ посредством программы структурно-семантического распознавания, которая является частью данного комплекса и выполняет полуавтоматическое распознавание информационно-справочных таблиц в составе документов Microsoft Word, преобразование их в ИЛТ и занесение в базу ресурсов комплекса. При этом все действия по импорту таблиц из документов Microsoft Word выполняются только кликами мыши.

Изложенная выше функциональность СИЛТ показывает целесообразность его применения в любых программах автоматизированного проектирования, в которых используется значительное количество нормативно-справочной информации из удаленных источников.

6. Создание баз знаний посредством СИЛТ

Известно, что во многих областях знаний принятие решений или достижение некоторой цели осуществляется посредством деревьев решений, вершины (узлы) которых являются состояниями из определенного множества состояний.

Напомним, что ИЛТ [4] может представлять собой логическую операцию по выбору решения из ограниченного множества возможных решений. Если этими решениями будут являться некоторые имена состояний из множества всех возможных состояний (пространства состояний), то ИЛТ можно рассматривать как формальное представление локальной возможности перехода от одного состояния к другому, определяемой объективно существующими условиями.

Принимая во внимание, что, с одной стороны, ИЛТ может рассматриваться как дерево либо прямого, либо обратного вывода, а, с другой стороны, одна ИЛТ может вставляться в ячейку другой ИЛТ, ИЛТ можно использовать в качестве вершин (узлов) деревьев решений.

На рис. 7 приведен пример плоского дерева решений, узлы которого представлены информационно-логическими таблицами, вставленными в ячейки выбираемых значений.

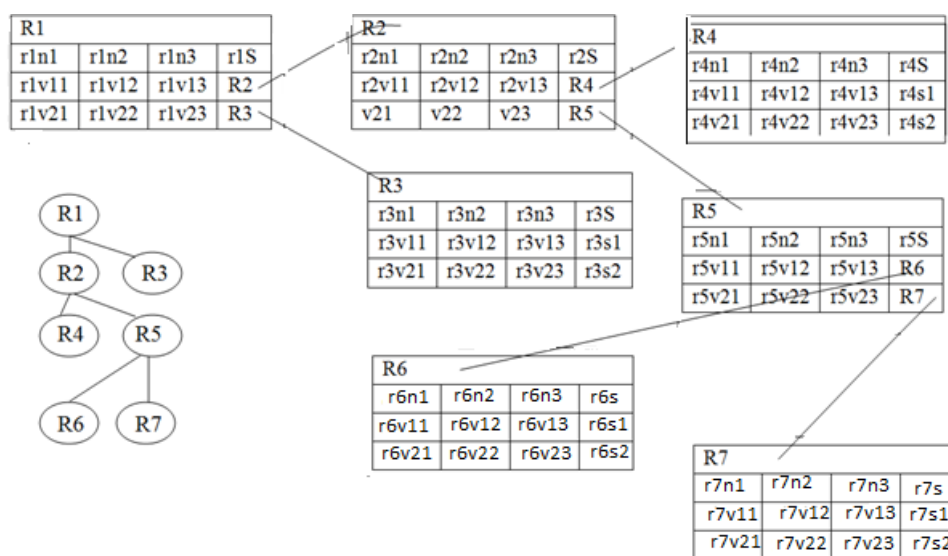


Рис. 7. Плоское дерево решений с узлами из информационно-логических таблиц

Вставка ИЛТ в промежуточную ячейку изменяет логику вывода (рис. 8). Для такой ИЛТ нужно выполнить обратный обход дерева, представленного вложенной ИЛТ, вернуться с решением в этот же узел родительской ИЛТ, после чего продолжить обход родительского дерева решений. Полученное решение становится значением факта, который используется в процессе выбора логической ветви. Дочерняя ИЛТ может, в свою очередь, содержать другие дочерние ИЛТ, которые также должны рассматриваться как деревья обратного обхода. В результате может быть сформировано пространственное дерево решений произвольной сложности (рис. 9).

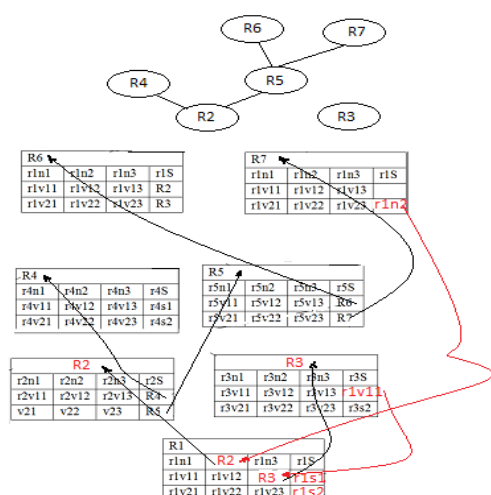


Рис. 8. Вложенные ИЛТ обратного обхода

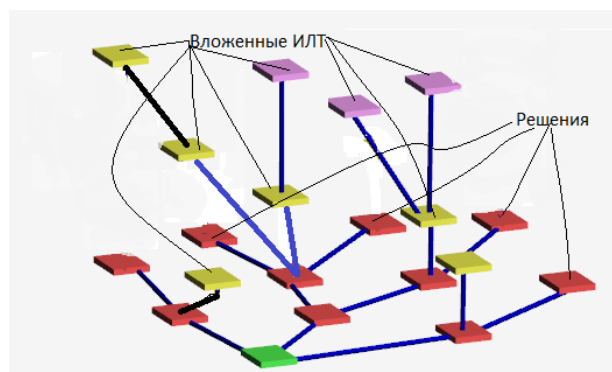


Рис. 9. Пространственное дерево решений на основе ИЛТ

7. Монитор СИЛТ

ИЛТ сама по себе, а также БИР, представляющая дерево решений, являются пассивными объектами. Вместе с тем интерпретация ИЛТ в виде группы правил «если – то» предполагает существование механизма, в рамках которого имена атрибутов получают значения из внешней среды, выполняется процесс вычисления логического выражения и поиск соответствующего решения. Реализацию этих функциональностей принимает на себя встроенная в СИЛТ программа-монитор, которая имитирует действия конечного пользователя и содержит два интерфейса:

- визуальный интерфейс взаимодействия с пользователем, в рамках которого выводятся имена ИЛТ, имена атрибутов и список допустимых значений каждого атрибута с просьбой сделать выбор значения из предлагаемого списка. В результате формируется набор входных параметров, определяющих выбор логической ветви на каждом узле дерева решений;
- программный интерфейс, посредством которого выполняются поиск связей и рекурсивная обработка каждой ИЛТ в цепочках связей.

Оба интерфейса работают параллельно.

Рассмотренные принцип и приемы создания моделей распределенных инженерных знаний разрабатывались на протяжении ряда последних лет на кафедре САПР БНТУ и были апробированы во время выполнения научно-исследовательской программы Союзного государства Беларуси и России «Исследовать, разработать и создать инфраструктуру подготовки специалистов для решения сложных прикладных задач на основе ВМВС» по заданию «Исследовать, разработать и создать инфраструктуру подготовки специалистов для решения сложных прикладных задач на основе ВМВС», а созданный при этом программно-информационный комплекс СИЛТ был внедрен на кафедре САПР БНТУ. В настоящее время он используется в качестве программного обеспечения дисциплины «Базы знаний и поддержка принятия решений в САПР», которая входит в государственный компонент учебного плана специальности 1-40 05 01-01 «Информационные системы и технологии (в проектировании и производстве)».

Заключение

В настоящей работе на основе принципа, который ограничивает трактовку термина «инженерные знания» как рекомендацию выполнять определенное действие в определенных условиях, предложены способ формализованного представления инженерных знаний в форме ИЛТ, универсальная логическая схема реляционной базы данных, обеспечивающая хранение

инженерных знаний, а также принцип объединения распределенных инженерных знаний в единое пространство знаний посредством создания кластера комплексов СИЛТ.

Изложена методика создания баз распределенных инженерных знаний посредством программно-информационного комплекса СИЛТ [4].

Список литературы

1. Автоматизированные системы технологической подготовки в машиностроении ; под ред. чл.-корр. АН БССР Г.К. Горанского. – М. : Машиностроение, 1977. – 240 с.
2. Евгенийев, Г.Б. Системология инженерных знаний : учеб. пособие для вузов. – М. : Изд-во МГТУ им. Н.Э. Баумана, 2001. – 376 с.
3. Интермех. Комплексная автоматизация предприятия [Электронный ресурс]. – Режим доступа : <http://www.intermech.ru>. – Дата доступа : 14.09.2017.
4. Базы знаний и поддержка принятия решений в САПР [Электронный ресурс]. – Режим доступа : <http://silt.bntu.by/>. – Дата доступа : 14.09.2017.

Поступила 15.09.2017

*Белорусский национальный
технический университет,
Минск, пр. Независимости, 65
e-mail: vakochurov@tut.by*

V.A. Kochurov

A METHOD OF CREATING DISTRIBUTED ENGINEERING KNOWLEDGE DATABASES

The article reviews the principle and methods for creating the models of distributed engineering knowledge as well as software implementation of such models in the form of the Information and Logical Table Server (SILT) resulted in development of a cluster based on the WCF (Windows Communication Foundation) technology and reciprocity of the SILT elements. The cluster might be entered by means of a special web application which is an element of SILT using every device with a web browser and access to the internet.