

УДК 628.440.22

А.В. Григорьев, О.В. Морозова

Донецкий национальный технический университет, г. Донецк
кафедра прикладной математики и информатики
E-mail: grigorie@r5.dgtu.donetsk.ua, malyaw0n@mail.ru

АНАЛИЗ ЭФФЕКТИВНОСТИ И ПЕРСПЕКТИВ РАЗВИТИЯ МЕТОДОВ ПОСТРОЕНИЯ ДВУХСТОРОННИХ ТРАНСЛЯТОРОВ В ЗАДАЧЕ СОЗДАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ НАДСТРОЕК НАД ПРОБЛЕМНО- ОРИЕНТИРОВАННЫМИ САПР

Аннотация

Григорьев А.В., Морозова О.В. Анализ эффективности и перспектив развития методов построения двухсторонних трансляторов в задаче создания интеллектуальных надстроек над проблемно-ориентированными САПР. В работе обсуждается эффективность метода построения двухсторонних трансляторов в задаче построения интеллектуальных надстроек над проблемно-ориентированными САПР в рамках инструментальной оболочки для построения интеллектуальных САПР, а так же пути повышения эффективности данного процесса. Задача решается в условиях специфики, включая физическую семантику языков, уровень абстракции языков, уровень квалификации эксперта в предметной области. Обсуждается ряд реализаций таких интерфейсов для различных САПР.

Ключевые слова: физическая семантика предметной области, проблемно-независимый язык спецификаций, язык проблемно-ориентированного САПР, инструментальная оболочка для построения интеллектуальных САПР, интеллектуальная надстройка на САПР, методы построения двойных трансляторов.

Общая постановка проблемы.

На пути автоматизации построения интеллектуальных САПР, ориентированных на заданную предметную область, возникает ряд новых задач, либо ранее еще не ставившихся и требующих новых методов решения, либо ставившихся, но не имеющих какого-либо общего метода решения, либо – имеющих недостаточно общий метод решения. Соответственно, такое положение делает актуальным либо построение, либо развитие подобных методов. Примером такой задачи является задача построения двухсторонних трансляторов в рамках создания интеллектуальных надстроек над проблемно-ориентированными САПР (далее – П САПР). Типичный современный П САПР, не зависимо от предметной области, представляет собой комплекс, обладающим такими компонентами: 1) специализированным графическим редактором, обеспечивающим пользователю интерфейс для создания и редактирования графических образов требуемого изделия на базе специфического набора команд и набора тех или иных примитивов; 2) базой данных, локальной или распределенной; 3) интерфейсами, как со смежными САПР, так и с системами производства изделий на основе разработанного проекта; 4) набором расчетных модулей, позволяющих управлять вводом, расчетом и корректировкой параметров, как отдельных компонента, так и проекта в целом; 5) системой моделирования, позволяющей ставить эксперименты над моделью для ее всестороннего исследования; 6) средствами отображения готовых изделий в форме чертежа; 7) средствами построения смет, спецификаций и т.д. К таким САПР можно отнести OrCAD, ArchiCAD т.д. Данные пакеты можно было бы назвать «классическими» в том плане, что в них практически отсутствуют или в незначительной мере используются методы искусственного интеллекта и, в частности, базы знаний. Однако, главная современная тенденция повышения эффективности

функціонування САПР в нинішнє час – це підвищення рівня їх інтелектуальності, т.е. використання сучасних високоєфективних методів: 1) створення продукційної бази знань, що дозволяє автоматизувати ті чи інші процеси проектування; 2) виконання розрахунків, включаючи нейронні мережі, різноманітні методи програмування в обмеженнях; 3) винаходи нових рішень на основі генетических алгоритмів тощо.

Прогресивні САПР будуються вже як гібридні, т.е. як включаючі «інтелектуальні» методи побудови, моделі і засоби поряд з «класическими». Задачею гібридних САПР є більш ефективне відображення методик проектування об'єктів, існуючих в даній предметній області. Якщо існуючий САПР не в повній мірі реалізує існуючу методологію проектування пристроїв в даній предметній області, то можна підвищити рівень його ефективності, надстраиваючи над ним базу знань для синтезу (редагування, моделювання, документування) рішень по вимогній методології проектування. Не менш важливою тенденцією сучасних інтелектуальних засобів для створення експертних систем є тенденція виключення з процесу створення експертної системи в межах певної інструментальної оболонки (ІО) інженера по знанням на користь експерта в предметній області, в даному випадку – проектувальника – користувача САПР. В цьому випадку важливо, щоб: 1) інструментальні засоби по семантиці і по формі побудови були доступні експерту в предметній області; 2) «готові» інтелектуальні компоненти були орієнтовані на рівень підготовки типового проектувальника. Так як експерти в предметній області можуть мати різну підготовку, то передбачається орієнтація ІО на рівень кваліфікації експерта в предметній області. Можливо виділити рівні умовної градуювання експертів [1]: 1) «умний» експерт, здатний явно задати методологію проектування в формі продукційної бази знань; 2) «середній» експерт, здатний задати основні складові частини методик проектування, що дозволяє ІО побудувати відповідну базу знань в напівавтоматичному режимі; 3) «глупий» експерт, не здатний допомогти ІО в задачі побудови БЗ, але здатний надати «достовірні» рішення, що змушує ІО побудувати відповідну базу знань в автоматичному режимі.

Можливо різні шляхи гібридування вже існуючих САПР: 1) включення авторами-розробниками даного САПР нових інтелектуальних «закритих» компонентів, що підвищують його функціонування; 2) забезпечення користувачу-проектувальнику можливість самому будувати нові інтелектуальні компоненти, минувши розробників. Другим шляхом передбачається побудова інтелектуальної надстройки над САПР, що розширює його можливості, самим інженером-проектувальником. В цілому ж можливо виділити основні параметри адаптації ІО для проблемної області: 1) Специфіка методик проектування в даній предметній області; 2) Рівень існуючого втілення методик проектування в формі П САПР; 3) Рівень здатності експерта в предметній області, що грає роль інженера по знанням, явно сформулювати свої знання по методології проектування з метою підвищення ефективності рівня втілення методик проектування. Так, наприклад, висококваліфікований (умний) експерт в предметній області на основі детального вивчення методик проектування, описаних авторами методик, і широкого досвіду розробки подібних пристроїв різного типу і призначення, вміщує в собі експерта в предметній області і інженера по знанням. Це призводить до виключення інженера по знанням з ланки розробки програмного продукту. Для теоретичного рішення проблеми побудови інтелектуальних САПР в середовищі спеціалізованих ІО, в тому числі – і побудови інтелектуальних надстроек над П САПР, авторами було розроблено апарат концептуальної моделі предметної області (далі – КМ П) [2-6], що включає як компонент метод побудови двонаправлених (двійних) трансляторів. Основу КМ П становить система взаємопов'язаних проблемно-незалежних мов специфікацій (ЯС) різних рівнів абстракції і повноти, які можуть відображати специфіку вимогній предметній області. Кожен ЯС становить основу побудови бази знань (БЗ) в середовищі ІО, що відображає методологію проектування в тій чи іншій предметній області на тому чи іншому рівні абстракції з урахуванням рівня кваліфікації експерта в

предметной области. БЗ как совокупность модулей знаний отражает структуру модели объекта проектирования. Основой представления знаний в модулях знаний является грамматика, эквивалентная И-ИЛИ-дереву, с определенными над ней продукционными зависимостями. На основе данной КМ П авторами был построен ряд оболочек, а также создан в их среде ряд интеллектуальных надстроек над П САПР, включая надстройки над: САПР экономических расчетов – Excel [7]; САПР бизнес-планов – ProjctExpert [8]; САПР вычислительной техники – VHDL [9]; САПР парогазовых установок – СПРУТ [10]. Перечисленные надстройки отличаются решаемой задачей, полнотой возможностей, ориентацией на тот или иной уровень квалификации эксперта в предметной области, а так же - уровень абстрактного представления модели (структурный, функционально-логический, количественный) и несут в себе все достоинства и недостатки тех или иных версий инструментальных оболочек. Еще ранее ряд положений, положенных в основу данной КМ П, был апробирован при построении инструментальной оболочки ИнтерНаМ, интеллектуального САПР микропроцессорных систем – EMULAT, а также САПРов программного обеспечения RPG/M и R-тран.

Задача построения интеллектуальной надстройки для данного П САПР, независимо от ориентации, предполагает наличие двух языков: 1) проблемно-независимого языка формальных спецификаций (далее - ЯС), имеющего физическую семантику, составляющего основу представления знаний в ИО; 2) конкретный внутренний язык П САПР, используемый для описания модели объекта проектирования. Результирующая модель объекта проектирования, синтезированная в среде оболочки на основе соответствующего языка задания технического задания, должна быть передана в среду П САПР для решения задач: графического отображения; исследование модели; документирования полученных результатов и т.д. При этом этап ручного построения модели объекта проектирования в среде графического редактора П САПР – выпадает. Тем самым повышается эффективность процесса функционирования П САПР. Но в данном случае возникает необходимо выполнить трансляцию «ЯС -> язык П САПР». В зависимости от специфики метода построения БЗ, соответствующего уровню квалификации пользователя, специфики задачи, решаемой в САПР, необходимой может быть и обратная задача, а именно – задача обратной трансляции «язык П САПР -> ЯС». Такая задача возникает, в частности, при: 1) построении БЗ путем обучения на основе прототипов решений, полученных ранее в П САПР; 2) решении задачи реконструкции объекта проектирования, при которой необходимы как перевод модели на языке П САПР в ЯС, изменения ее в данной форме представления, а так же последующая передача измененной модели в П САПР.

Таким образом, задача построения интеллектуальных надстроек требует создания двухсторонних трансляторов в паре «ЯС – язык П САПР». Задача построения трансляторов, естественно, может быть решена с использованием известных типичных методов [11] и средств программирования, например, системы FLEX. Однако, данная задача имеет специфику, связанную с: 1) Семантикой связываемых языков, т.е.: физической семантикой языка П САПР некоторого уровня абстракции; физической семантикой ЯС соответствующего уровня абстракции; 2) Формами, доступными для передачи или получения данных о структуре объекта проектирования в П САПР: грамматикой языков представления моделей предметной области в П САПР; форматом внутренних структур данных представления моделей предметной области в П САПР; 3) Формами, доступными для передачи или получения данных о структуре объекта проектирования в ИО: грамматикой ЯС соответствующего уровня абстракции; форматом внутренних структур данных представления моделей предметной области в ИО.

Т.о., имеется специфика задачи построения таких трансляторов. Решение задачи построения такого рода интерфейсов с учетом данной специфики в современной литературе практически отсутствует. Так, с точки зрения теории Data Mining речь идет о специфической задаче извлечения информации из формальных текстов или баз данных, пока не имеющей аналога. Можно назвать для примера, близкие, но не эквивалентные методы:

1) Теория физических структур, совмещенная с теорией измерений [12];
достоинства: достаточно обоснованная попытка отразить законы природы; *недостатки*:

фактически нет физической семантики, как она понимается в САПР (потенциалы, потоки и т.д.) и не рассматривается уровни абстракции языков; слабо проработаны алгоритмы извлечения, соответствующие специфике подхода;

2) Методы машинного обучения в задачах извлечения информации из текстов [13], включая:

- методы символического подхода, где правила извлечения описываются языками, близкими к регулярным выражениям, при этом обучение в таких методах строится либо по принципу дедуктивного вывода, либо по принципу индуктивного обобщения; пропозиционные и реляционные методы эквивалентны, соответственно, логике нулевого и - первого порядка; на обучающей выборке, исходя из выбранного метода обучения, синтезируются обобщенные правила распознавания целевой информации; *достоинства*: индуктивный принцип обобщения, грамматический подход; *недостатки*: не рассматривается физическая семантика текстовых компонент;

- методы численного подхода, где извлекатель строится как подбор комбинации изначально заданных правил, исходя из заданных уровней точности и полноты извлечения; *достоинства*: возможность учитывать наличие известных правил, по которым строится текст; *недостатки*: не рассматривается физическая семантика текстовых компонент и т.д.

Т.о., речь идет о новой задаче, ранее еще не ставившейся и требующей новых методов решения. Ранее авторами была предложена общая методика построения данных трансляторов (интерфейсов), позволяющая учитывать специфику задачи [7]. В рамках данной модели и выполнялось решение задач построения интерфейсов для тех или иных П САПР.

Постановка задач исследования.

Исходя из накопленного опыта построения таких интерфейсов, актуальным становится решение двух задач: 1) задачи анализа эффективности построения двухсторонних трансляторов в процессе создания интеллектуальных надстроек над П САПР на основе предложенных общих подходов; 2) задача анализа проблематики развития предложенных подходов.

Цель работы: 1) Описать используемый общий метод построения двойных трансляторов; 2) Выявить специфику каждой задачи построения двойного транслятора для конкретных П САПР; 3) Выполнить детальное рассмотрение решения типичной конкретной задачи построения двойного транслятора для одного П САПР с целью показать все проблемы этого процесса; 4) Путем анализа имеющегося опыта, выполнить оценку предложенного общего метода построения таких интерфейсов на эффективность, полноту и общность; 5) Выявить перспективные направления развития предложенного общего метода построения интерфейсов с целью повышения эффективности процессов построения таких интерфейсов. Т.о., речь идет о том, что бы исследовать, достаточно ли эффективен предложенный метод решения, а так же - определить пути его развития.

Решение задач и результаты исследований.

Выполним решение поставленных выше задач.

1. Методика организации интерфейса двухстороннего транслятора

Для разработки методов построения транслятора необходимо рассмотреть принципы адаптации ЯС к особенностям языка П САПР. Принципы адаптации: 1) уровень абстрактного представления языка П САПР определяет соответствующий уровень абстрактного представления ЯС; 2) семантика языка П САПР приводится в соответствие к семантике ЯС; 3) грамматика ЯС (идентификаторы компонент модели) отражает грамматику языка П САПР. Сама задача адаптации возникает и решается, исходя из квалификации эксперта в предметной области, управляющего созданием надстройки: 1) умный эксперт – предполагает обязательными все этапы адаптации; 2) средний эксперт – может предполагать адаптацию на семантику, а может и – нет; в этом случае используется автоматический

механизм такой адаптации, без учета семантики; 3) глупый эксперт – может не предполагать такой адаптации, точнее - адаптация может делаться автоматически, без учета семантики.

Суть методики адаптации – это приведение семантики П-языка к семантике формального ЯС соответствующего уровня абстракции в условиях особенностей семантики ЯС. Такой путь подразумевает: 1) Выявление в П-языке семантики предполагаемой модели пространства и времени; 2) Выявление в П-языке фазовых переменных, т.е. потенциала и потока, координат взаимодействия, топологических свойств, а так же приведение их к элементам ЯС на основе принципа аналогии; 3) Выявление в модели структурного блока П-языка компонент модельной структуры языка спецификаций (внешняя и внутренняя границы, связи внутренние и внешние, соответственно, чужие и свои); 4) Выявление в П-языке модели функций базовых структурных блоков; 5) Выявление в П-языке модели функций небазовых функциональных блоков различных уровней иерархии. Требуемая характеристика внутреннего языка (структур данных) П САПР: наличие физической семантики П; отсутствие нечеткостей, неоднозначностей, альтернативности в описании объекта проектирования некоторой известной структуры.

Модель предлагаемого интерфейса может быть представлена следующим образом:

$$M=(Go,Fo,Mo,So,Gs,Fs,Ms,Ss,Pos,Pso),$$

где: Go - грамматики языков представления моделей в ИО; Fo - формат внутренних структур данных представления моделей в ИО; Mo - описания прототипов в данной проблемной области на языках ИО (знания экспертов, проекты); So - описания прототипов в данной проблемной области в формате внутренних структур данных МЭО; Gs - грамматики языков представления моделей в других инструментальных средствах проектирования сложных систем (САПР); Fs - внутренние структуры данных представления моделей других САПР в данной проблемной области; Ms - описания прототипов в данной проблемной области на языках прочих САПР (знания экспертов, проекты); Ss - описания прототипов в данной проблемной области в формате внутренних структур данных прочих САПР; Pos - процедуры отображения Mo,So в Ms,Ss; Pso - процедуры отображения Ms,Ss в Mo,So. Охарактеризуем формы представления моделей в ИО: 1) Грамматика языка представления моделей в ИО (Go) описана в [6]; 2) Модель структур во внутреннем формате (So), представляется в виде описаний библиотек, типов, массивов и т.п. в формате DBF; 3) Функциональные модели, задающие соответствия для базовых структурных блоков и связей, представляются в виде динамических недоопределенных вычислительных моделей, описание которых совмещено с описанием структур (So) и представлено в формате DBF.

2. Анализ эффективности решения задачи построения двойного транслятора для конкретного проблемно-ориентированного САПР, над которым строится интеллектуальная надстройка.

Рассмотрим специфику и эффективность процессов построения «двойных трансляторов для различных наиболее типичных П САПР.

2.1. Построение интерфейсов для САПР экономических расчетов – Excel

Задача: построить надстройку над Excel, которая, имея Excel-таблицу, решающую задачу в некоторой постановке, обеспечивает синтез новых таблиц, предназначенных для решения задачи в измененной постановке, т.е.: имея Excel-таблицу, предназначенную для расчетов ряда выходных параметров-ячеек по ряду входных параметров-ячеек, а так же – ряд промежуточных ячеек, обеспечить изменение постановки задачи, изменив роль всех или ряда ячеек, т.е. изменение входов - на выходы, а так же промежуточных ячеек – на входные или выходные; цель – обеспечить для имеющейся таблицы решение задачи в любой постановке.

Специфика задачи: 1) ориентация ИО на эксперта средней или низкой квалификации, способного проанализировать Excel-таблицу, выявить в ней постановку задачи и – потребовать новой постановки задачи; 2) отсутствие в Excel широко известных форматов внутренних структур данных, доступных для анализа; 3) - наличие в Excel языка формул делает возможным рассмотрение его как языка П САПР, используемого в трансляторе «язык П САПР -> ЯС»; 4)

синтез решения в ИО состоит в выборе пользователем нового набора входных и выходных ячеек, задающих новую постановку задачи, а затем – автоматической модификации системы уравнений, ранее заданных формулами, что требует использования в ИО метода динамических вычислительных моделей с целью создать новый набор формул, связывающих данные в рамках требуемого порядка расчетов; 5) синтезированное решение переводится в язык формул Excel, что требует построение транслятора «ЯС -> язык П САПР».

Специфика способа решения задачи:

1) Физическая семантика языка П САПР, приведенная к специфике физической семантики ЯС:

- уровень абстракции – количественная макромодель, при том, что топологические свойства блоков отражают адресацию ячеек в таблице;

- таблица Excel представляет собой статическую модель, т.е. – единственную модель пространства в жизненном цикле, т.е. – модель времени фактически отсутствует;

- структура Excel-таблицы, где ряд ячеек объединен по данным, позволяет рассматривать их как подблоки, т.е. отдельные физические точки в модели пространства, входящий в общий блок – «модель расчетов»;

- направление передачи данных между подблоками: входы блоков - это идентификаторы «чужих» данных в формуле, записанной в ячейке; выход ячейки-блока – ее собственный идентификатор;

- иерархическая структура блоков – не рассматривалась, но косвенно учитывалась в существующем формате таблиц.

2) Механизм реализации интерфейса: система LEX.

Возможности надстройки как БЗ: на выходе – новая Excel-таблица, решающая задачу расчетов в новой постановке.

Роль Excel как САПР-приемника для синтезированных решений: - моделирование – т.е. решение задачи в новой постановке; документирование - т.е. построение «красивых» таблиц и графиков.

2.2. Построение интерфейсов для САПР бизнес-планов – ProjectExpert

Задача: обучить ИО на примере готовых бизнес-планов с целью построить надстройку, позволяющую выбирать готовый бизнес-план, наиболее подходящий конечному пользователю-экономисту.

Специфика задачи: 1) ориентация ИО на эксперта низкой квалификации, что обуславливает необходимость передачи решения – бизнес плана из среды ProjectExpert целиком как прототипа для обучения базы знаний ИО; 2) отсутствие в ProjectExpert широко известных форматов внутренних структур данных, доступных для анализа; 3) наличие в ProjectExpert возможности экспортировать вовне готовые бизнес-планы, включая таблицы денежных потоков, в формате языка HTML, что делает возможным рассмотрение HTML как языка П САПР, используемого в трансляторе «язык П САПР -> ЯС»; 4) синтез решения в ИО состоит в выборе готового бизнес-плана, подходящего пользователю, в форме идентификатора файла готового бизнес-плана, что делает излишним построение транслятора «ЯС -> язык П САПР», поскольку требуемое решение в формате «языка П САПР» уже имеет место.

Специфика способа решения задачи:

1) Физическая семантика языка П САПР, приведенная к специфике физической семантики ЯС:

- уровень абстракции – количественная макромодель, при том, что топологические свойства блоков отражает адресация ячеек в таблице;

- таблица денежных потоков, где каждый столбец соответствует некоторому периоду времени, рассматривается как жизненный цикл, т.е. – модель времени;

- структура таблицы, где ряд смежных строк объединены одной операцией (расчет убытков, расчет прибыли и т.д.), позволяет рассматривать их как единый блок, как подблок входящий в общий блок – «модель предприятия», отдельные строки трактуются как подблоки самого нижнего уровня;

- направление передачи данных между подблоками – сверху-вниз;
- совокупность подблоков, входящих в иерархию, и связи между ними – это модель пространства;

- идентификаторы строк – это идентификаторы блоков и их выходных данных, группа строк идентифицируется итоговой (нижней) строкой; идентификатор столбца – идентификатор момента времени в жизненном цикле;

2) Механизм реализации интерфейса: система FLEX.

Возможности надстройки как БЗ:

- обобщение ЖЦ «по вертикали» [8] позволяет создавать БЗ, способную обеспечить задание пользователем требований типа: «производство должно начаться ранее 2 года», «интегральный экономический эффект на 5 году более 25%» и т.д.;

- на выходе – один или группа бизнес-планов, отвечающих требованиям пользователя.

Роль ProjectExpert как САПР-приемника для синтезированных решений: - модификация выбранных решений с целью приблизить полученный прототип к желаемому решению.

2.3. Построение интерфейсов для САПР парогазовых установок – СПРУТ

Задача: создать в среде ИО надстройку на САПР парогазовых установок (ПГУ) в энергетике СПРУТ, способную решать как задачу синтеза по некоторой методике проектирования требуемой ПГУ с последующей передачей решения в САПР СПРУТ, так и – задачу реконструкции существующего решения ПГУ по некоторой методике реконструкции, вначале – получив его из среды СПРУТ, а затем – передав измененное решение в СПРУТ.

Специфика задачи: 1) ориентация ИО на эксперта высокой квалификации, способного описать в виде набора продукций методику синтеза или реконструкции ПГУ; продукции ориентированы на использование конструкций на ЯС в посылках и выводах продукций; 2) наличие в СПРУТ известных авторам (как разработчикам СПРУТ) форматов внутренних структур данных, доступных для анализа; фактически структуры данных, позволяющие описать структуру объекта проектирования, есть подмножество языка АвтоЛИСП; 3) реконструкция решения в ИО состоит в анализе имеющихся особенностей существующей ПГУ и выборе структурных особенностей для требуемой ПГУ, что обуславливает необходимость передачи решения – структуры ПГУ из среды СПРУТ в среду ИО целиком как прототипа, подвергаемого реконструкции; 4) синтез решения в ИО состоит в выборе структурных особенностей требуемой ПГУ в терминах ЯС, построения решения в виде файла на ЯС, что требует построения транслятора «ЯС -> язык П САПР» для получения решения в форме ЛИСП-файла.

Специфика способа решения задачи:

1) Физическая семантика языка П САПР, приведенная к специфике физической семантики ЯС:

- модель ПГУ рассматривается как статическая, т.е. – неизменная во времени, т.о. рассматривается одна модель пространства на жизненном цикле, т.е. фактически модель времени – отсутствует;

- модель ПГУ рассматривается как количественная макромоделю для подуровня макромоделю; т.е. – как модель с сосредоточенными параметрами, а фактически – как набор физических точек (ФТ), с трехмерными пространственными координатами и расширенным набором топологических и «обычных» свойств: длина, вес, материал и т.д.; параметры ФТ связаны алгебраическими зависимостями;

- модель структуры – иерархическая, т.е. ряд ФТ объединяются в блоки – трубу некоторого типа, оборудование, стену и т.д.; трубы объединяются в трубопровод, рассматриваемый как массив труб и т.д.; рассматривают точки стыков – как функции взаимосвязи двух или более блоков;

- направление передачи данных между подблоками – двунаправленное;

- совокупность подблоков, входящих в иерархию, составляющую модель ПГУ, а так же связи между ними – это модель пространства;

- идентификаторы блоков, массивов, номеров блоков в массиве – явно заданы в ЛИСП-файле;

2) Механизм реализации интерфейса: система FLEX.

Возможности надстройки как БЗ: на выходе – структура ПГУ, удовлетворяющая заданным требованиям.

Роль СПРУТ как САПР-приемника для синтезированных решений: моделирование – т.е. доопределение ряда неопределенных параметров решений, поиск конфликтов и т.д.; документирование - т.е. формирование чертежей и спецификаций для новых или модифицированных решений.

3. Детальное рассмотрение процесса построения двустороннего транслятора для САПР СПРУТ

С целью более детального анализа проблем построения интерфейса опишем процесс построения двустороннего транслятора для САПР СПРУТ. Модель предлагаемого двустороннего транслятора может быть представлена описанием пары трансляторов М и М'. Специфика данного интерфейса: 1) семантика языка П САПР практически полностью совпадает с семантикой ЯС соответствующего уровня абстракции; 2) грамматика двух языков существенно отличается. Модель транслятора М в ЯС может быть представлена выражением:

$M=(Go,Fo,Mo,So,Gs,Fs,Ms,Ss,Pos,Pso)$, где:

Go - грамматики языков представления моделей предметной области; Fo - формат внутренних структур данных представления моделей предметной области – база знаний (БЗ); Mo - описания прототипов (элементарных элементов) в данной проблемной области на языках проблемно-ориентированных САПР (знания экспертов, проекты); So - описания прототипов в данной проблемной области в формате внутренних структур данных; Gs - грамматики языков представления моделей, независимых от предметной области; Fs - внутренние структуры данных представления моделей, независимых от предметной области; Ms - описания прототипов в данной проблемной области на проблемно-независимых языках (знания экспертов, проекты); Ss - описания прототипов в данной проблемной области в формате внутренних структур данных; Pos - процедуры отображения Mo,So в Ms,Ss; Pso - процедуры отображения Ms,Ss в Mo,So. Охарактеризуем формы представления моделей: 1) Грамматика языка представления моделей (Go) в предметной области представлена в форме Бэкуса—Наура и приведена на рис. 1. 2) Модель структур во внутреннем формате (So), представляется в виде описаний библиотек, типов, массивов и т.п. в формате базы данных в виде взаимосвязанных таблиц. 3) Функциональные модели, задающие соответствия для базовых структурных блоков и связей, представляются в виде моделей, описание которых совмещено с описанием структур (So) и представлено в формате базы данных. Особенности форм представления моделей: 1) Внешний язык описания моделей (Gs) может рассматриваться как входные продукции; 2) Имеется ряд внутренних форматов данных, предназначенных для экспорта-импорта данных в другие программные системы. Эти форматы представляют собой простой и наглядный язык текстовых представлений модели (Gs). Для создания транслятора $Go \leftrightarrow Gs(So)$ требуется выполнение подпроцедур: 1) определение пользователем грамматики языка описания исходных текстов (Gs); 2) семантический анализ грамматики Gs на выявление в ней элементов концептуальной модели, представленной описанием представленным грамматикой Go (или форматом So). Семантический анализ - выделение смысла высказывания и его представление на внутреннем языке системы. Семантический анализ производится над входными данными системы. В данном случае входными данными является входная грамматика. Грамматика Go, которая поступает на вход транслятора, переходит в блок лексического анализа, который, в свою очередь, состоит из двух подблоков - лексического и синтаксического анализа, предназначенных для анализа поступивших данных. Затем данные попадают в блок семантического анализа, который состоит из подблоков выделения стандартных подструктур и формирования внутреннего языка Gs. В данном блоке поступившие данные анализируются

со смысловой точки зрения, то есть семантический анализ приводит к выделению структур и формированию модели описания Ms.

```

<входное выражение> ::= <вид выражения> <входной предикат> <знак следствия>
<следствие> | <входной предикат> <знак связи> <следствие>
<вид выражения> ::= <существующее> | <вопросительное>
<входной предикат> ::= <название библиотеки> <разделительный знак> <тип>
<разделительный знак> <массив типов> <разделительный знак> <подтип> <разделительный
знак> <разделительный знак> <свойства блока> | <название библиотеки> <разделительный
знак> <тип> <разделительный знак> <массив типов> <разделительный знак> <свойства
блока>
<знак связи> ::= -
<знак следствия> ::= ->
<следствие> ::= <входной предикат>
<название библиотеки> ::= Труба | Арматура | Оборудование
<разделительный знак> ::= . | : | ( | ) | , | « | »
<тип> ::= прямая | уголок
<массив типов> ::= <название массива> [<число>] | <название массива> [<число> :
<число>]
<подтип> ::= <название массива> [<число>] | <название массива> [<число> : <число>]
<название массива> ::= <слово> | <символ>
<свойства блока> ::= <название свойства> = « <значение> »
<название свойства> ::= <слово> | <символ>
<значение> ::= <слово> | <число>
<слово> ::= <символ>
<существующее> ::= {~}
<вопросительное> ::= ?
<символ> ::= A | B | В | Г | Д | ... э | ю | я
<число> ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0
    
```

Рисунок 1 - Грамматика языка предметной области в ФБН

Следующим этапом является описание проанализированных моделей Ms. В данном блоке происходит создание целостного прототипа Ss предметной области. Получившийся прототип Ss предметной области поступает в блок механизма логического вывода и создает из него основные элементы для базы данных. Результатом процедуры Pos является отображения Mo,So в Ms,Ss. На выходе транслятора получается проблемно- независимый язык, который преобразуется в модель на языке Лисп. Модель транслятора M' в язык Лисп может быть представлена выражением:

$$M' = (Gs, Fs, Ms, Ss, Gl, Fl, Ml, Sl, Psl, Pls), \text{ где:}$$

Gs - грамматики языков представления моделей независимых от предметной области; Fs - внутренние структуры данных представления моделей независимых от предметной области; Ms - описания прототипов в данной проблемной области на проблемно-независимых языках (знания экспертов, проекты); Ss - описания прототипов в данной проблемной области в формате внутренних структур; Gl - грамматики языков представления моделей в языке Лисп; Fl - формат внутренних структур данных представления моделей предметной области в языке Лисп; Ml - описания прототипов (элементарных элементов) в данной проблемной области в языке Лисп; Sl - описания прототипов в данной проблемной области в формате внутренних в языке Лисп; Psl - процедуры отображения Ms,Ss в Ml,Sl; Pls - процедуры отображения Ml,Sl в Ms,Ss. Работа двухстороннего транслятора реализована в отображении процедур: 1) Pos - процедуры отображения Mo,So в Ms,Ss; 2) Pso - процедуры отображения Ms,Ss в Mo,So; 3) Psl - процедуры отображения Ms,Ss в Ml,Sl; 4) Pls процедуры отображения Ml,Sl в Ms,Ss.

Общий вид работы транслятора приведен на рис. 2.

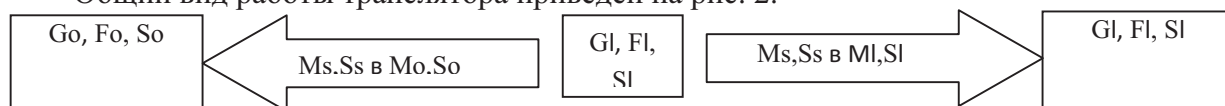


Рисунок 2 - Работа двухстороннего транслятора

При сворачивании модели можно получить входной текст в виде продукции типа: «Посылка -> Следствие». Входная продукция имеет семантику «если элемент существует» «то должен существовать следующий элемент». То есть, при выполнении условия, заданного в посылке некоторой продукцией, как существование некоторого элемента в результате предыдущего вывода, по следствию («то») обязательно должен существовать (не существовать) некоторый другой требуемый элемент. Например, если данный элемент не существует, то он должен быть добавлен в результат вывода. При решении задач синтеза или реконструкции во входном тексте ищется посылка и следствие. Посылка - это часть входного текста, который уже существует или не существует, через которую можно сделать привязку к следствию, к существующим элементам базы данных и создание новых.

В данном случае посылка представляет собой:

<Библиотека>.<Тип_библиотеки>.<Массив>[<Номер элемента>]:T1=«X,Y,Z»,..., TN=«X,Y,Z», диаметр = «значение»

Для дальнейшей работы интеллектуальной надстройки используется инструментальный пакет FLEX. Данный инструментальный пакет выделяет из набора символов - лексически значимые единицы. FLEX - это генератор программ, предназначенный для лексической обработки символьных входных данных. Он принимает проблемно-ориентированную спецификацию на высоком уровне и формирует программу на языке СИ, которая работает с регулярными выражениями. Регулярные выражения определяются пользователем в ключевой спецификации, выдаваемой программе lex. Система lex распознает эти выражения и разделяет входные данные на блоки в соответствии с ними. На границах между блоками исполняются фрагменты программ, разработанные пользователем, исходный файл lex объединяет регулярные выражения и фрагменты программ. По мере того, как выражения появляются на входе программы, составленной lex, соответствующий фрагмент подается на выполнение. Т.о., используется общепринятый подход к решению подобных задач, обеспечивающий высокую эффективность процессов обработки данных. В основе FLEX лежат этапы решения задач, начинающиеся с написания скрипта и далее: 1) скрипт попадает в лексический анализатор FLEX ++; 2) после обработки получается программный код лексического анализатора на языке С; 3) обработка кода на языке С в С++ компилятор; 4) создается программа лексического анализатора, которая выполняет обработку по заданному скрипту. Предлагается следующий скрипт грамматического разбора формального языка спецификаций, приведенный на рис. 3.

```
{ \. , \ " "->" \- : \[ \] = \? \( \) ~ }
```

Рисунок 3 – Скрипт для FLEX-анализатора

Вышеописанная посылка при попадании в лексический анализатор будет разобрана на лексемы. Над посылкой выполняется действие соответствующего скрипта. Кратко опишем методику работы интерфейса на этапе выделения лексем по заданному скрипту лексическому анализатору. При первом проходе анализатор разбивает входной текст на отдельные лексемы, проходя от начала строки до разделяющего символа "->". Он разделяет входной текст на посылку и следствие. В данном случае происходит поиск знаков, заданных во входной грамматике, которые отделяют один блок данных, т.е. одну лексему от другой. В нашем случае это знак ".", используемый для выявления лексем библиотеки, типа библиотеки и массива типов. Когда определяется лексема массива типов, так же находится знак "[" и "]", что означает перечисление элементов, входящих в массив типов, или количество вхождений в него, в зависимости от семантики. Следующим символом, по которому происходит выявление лексем, является знак ",", Он отделяет перечисляемые параметры один от другого. Знак ":" используется для разделения блока так называемых описаний и блока свойств этих описаний. Знак "=" присваивает параметру свойства значение, которые записываются после знака "<". Они являются характеристиками свойств и

в свою очередь могут быть перечисляемыми, т.е. разделяемыми знаком «,». Данный анализатор построен на четко сформулированном входном тексте. Описывается структура данных, т.е. набор таблиц, в которые записываются разобранные лексемы языка спецификаций и описывающих текущее состояние модели объекта проектирования, представлены на рис. 4.

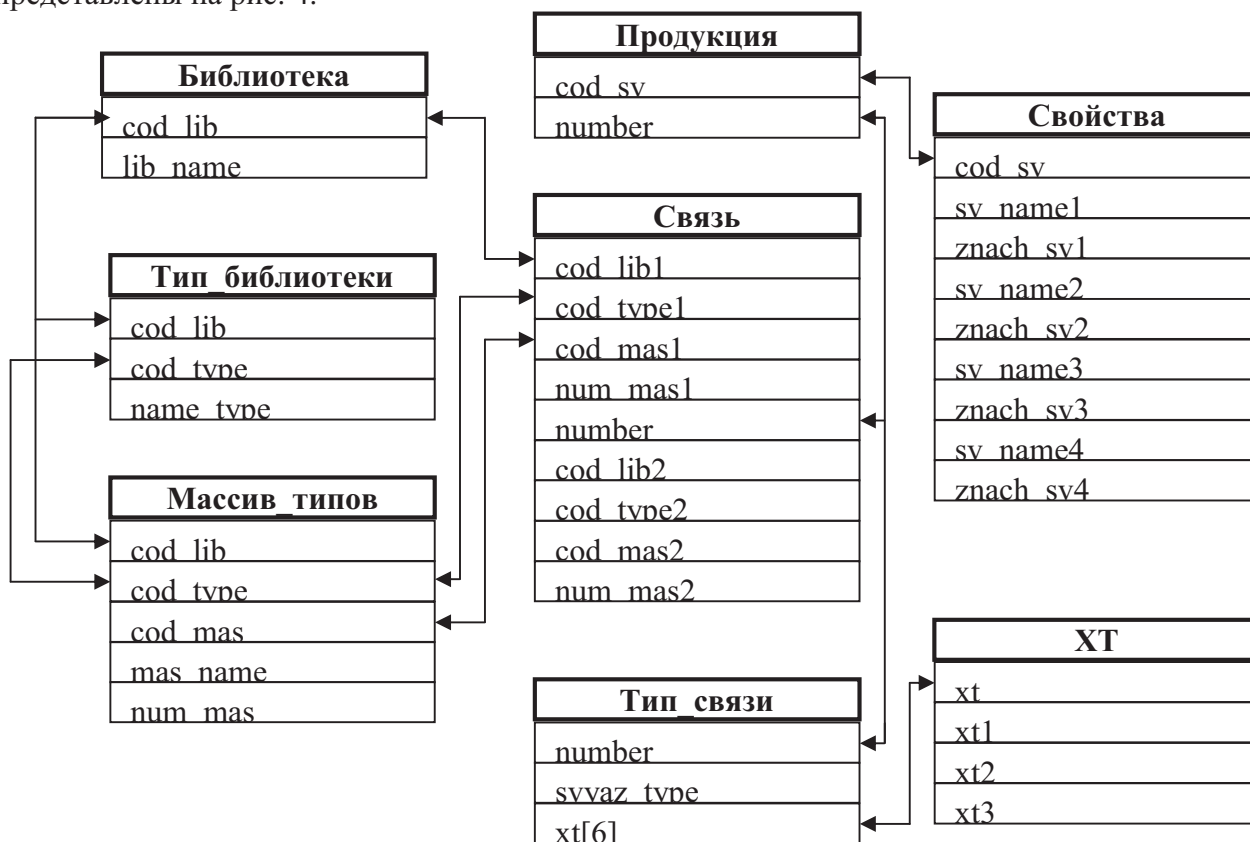


Рисунок 4 – Схема взаимосвязи структур данных

Для такого описания необходимо заполнить все поля таблиц. Некоторые поля дублируются в нескольких таблицах, это сделано, чтобы получить полноту полученных данных и обеспечить их четкую идентификацию. В данной структуре базы данных имеются таблицы-справочники, которые позволяют преобразовывать грамматические выражения одного языка в грамматические выражения другого языка, имеющих идентичный смысл. Если в справочнике не существует необходимого грамматического выражения, то необходимое выражение дописывается в конец существующего сборника, и в дальнейшем будет использоваться, как значение справочника. В представленной базе данных справочниками являются: библиотека, тип библиотеки и массив типов. Предложенная структура базы оказалась достаточно эффективной для реализации поставленной задачи проектирования и реконструкции сложных объектов интеллектуальной надстройки над проблемно-ориентированными САПР. Для того, чтобы данный анализатор работал правильно, необходимо строгое соблюдение семантики и порядка следования блоков. Для этого предлагается создание программного комплекса отслеживания семантических и синтаксических ошибок. После такого анализа входная продукция уже представляет собой набор лексем, которые и записываются в базу данных. Каждая лексема входной продукции соответствует определенной лексеме языка проблемно-ориентированного Лисп, изображено на рис. 5. Алгоритм преобразования спецификаций в ЛИСП приведен на рис. 6. Исходя из функциональности языка ЛИСП, все элементы записываются в виде списков, но в соответствии с выделенными лексемами. Для ускорения работы транслятора был составлен скрип-фильтр, по которому происходит лексический анализ данных. Это дает значительное

ускорение при работе транслятора, так как работа выполняется быстрее из-за того, что анализатор проходит всю строчку за один проход, а другие анализаторы - проходят текст за несколько проходов и анализ занимает больше времени.

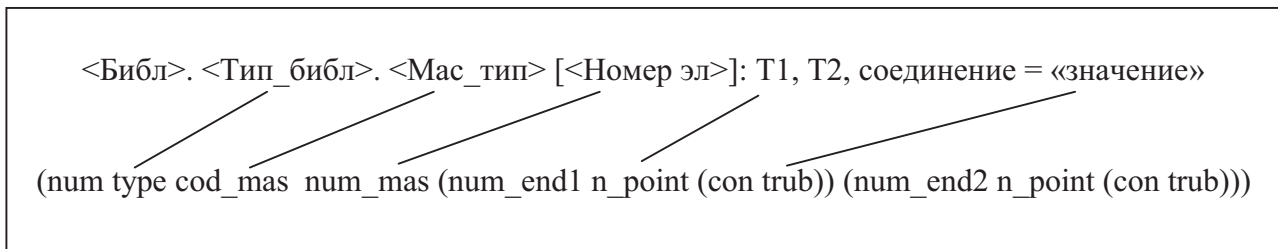


Рисунок 5 - Соответствие лексем входной грамматики и лексем Лиспа

```

запись строки «TRUB'S---»
запись количества труб трубопровода
для i от 1 до количества труб трубопровода {
  если (i = номер трубы продукции) то {
    запись номер трубы трубопровода
    запись кода типа трубы
    запись кода массива трубы
    запись номера элемента трубы
    запись номера конца трубы
    запись номера точки для конца трубы
    если (есть связь) то {
      запись кода типа связи
      запись номера следующей трубы
    }
    запись номера конца трубы
    запись номера точки для конца трубы
    если (есть связь) то {
      запись кода типа связи
      запись номера следующей трубы
    }
  }
}
    
```

Рисунок 6 - Алгоритм записи лексем Лиспа

Выводы.

В работе решены все задачи, поставленные целью работы. Прделанное исследование позволяет сделать следующие выводы: 1) Предложенная ранее общая методика построения «двойных» трансляторов (интерфейсов), позволяет в целом учитывать специфику решаемой задачи, что обеспечило решение ряда задач построения конкретных интерфейсов для тех или иных П САПР; 2) Многочисленные примеры с различной спецификой позволяют сделать вывод, что общая методика имеет достаточно высокую эффективность, полноту и общность; 3) Вместе с тем, имеются некоторые недостатки рассмотренной методики построения интерфейсов, связанные с недостаточной степенью автоматизации построения интерфейсов, что выражается в достаточно большой трудоемкости процессов их построения. В частности: 1) - каждый интерфейс строится индивидуально, хотя и на базе общих подходов; 2) - не существует единого программного средства, обеспечивающего построение двойного транслятора исходя из: специфики представления модели в П САПР (структура данных или язык), уровня квалификации эксперта, уровня абстракции языка П САПР, специфики физической семантики языка П САПР (модели пространства и времени, типа структурной иерархии и регулярности), полноты семантики языка П САПР относительно ЯС; 3) - мало используется положительный опыт ряда других эффективных подходов из известных методов Data Mining. Т.о., перспективной задачей является развитие предложенной методики, т.е. определение основных методов и принципов построения единого программного средства автоматизации построения «двойных» трансляторов для создания интеллектуальных надстроек над П САПР, учитывающего всю специфику данной задачи и использующего положительный опыт прочих методов Data Mining.

Литература

1. Григорьев А.В. Пути создания интеллектуальных САПР при различных уровнях квалификации экспертов / А.В. Григорьев // Научно-теоретический журнал «Искусственный интеллект». – 2005. - №3. – С. 758–763.
2. Григорьев А.В. Унифицированная концептуальная модель предметной области / А.В. Григорьев // В кн. Информатика, кибернетика и вычислительная техника (ИКВТ-97). Сборник трудов ДонГТУ. - 1997. - Вып. 1. - С.218-224.
3. Григорьев А.В. Семантика модели предметной области для интеллектуальных САПР / А.В. Григорьев // Научные труды Донецкого государственного университета: Серия «Информатика, кибернетика и вычислительная техника». – 2000. – Вып. 10. – С. 148-154.
4. Григорьев А.В. Комплекс моделей САПР как система взаимосвязанных уровней знаний о действительности / А.В. Григорьев // Научные труды Донецкого государственного университета: Серия «Информатика, кибернетика и вычислительная техника». – 2000. – Вып. 10. – С. 155-167.
5. Григорьев А.В. Теоретико-множественные операции над грамматиками как механизм работы со знаниями в интеллектуальных САПР / А.В. Григорьев // Вісник Східноукраїнського національного університету імені Володимира Даля. - 2002. - № 2(48). - С. 186-194.
6. Григорьев А.В. Специализированная оболочка для синтеза интеллектуальных САПР и АСНИ. (ИКВТ-97) / А.В. Григорьев, А.О. Базалей // Сборник трудов ДонГТУ. - 1997. - Вып. 1. - С. 225-228.
7. Григорьев А.В. Интерфейс табличного процессора EXCEL и специализированной оболочки для синтеза интеллектуальных САПР и АСНИ. (ИКВТ-97) / А.В. Григорьев, А.В. Бондаренко, А.В. Шойхеденко // Сборник трудов ДонГТУ. – 1997. - Вып. 1. - С. 229-238.
8. Григорьев А.В. Организация пространственного и временного логического вывода в концептуальной модели интеллектуальных САПР / А.В. Григорьев // Наукові праці Донецького національного технічного університету. Серія Інформатика, кібернетика та обчислювальна техніка (КОТ-2008). - 2008. - Вип. 9(132).– С.296-311.
9. Григорьев А.В. Интеллектуализация процесса проектирования аппаратуры средствами языка VHDL / А.В. Григорьев, Д.А. Кошелева // Наукові праці Донецького національного технічного університету. Серія «Інформатика, кібернетика та обчислювальна техніка та автоматизація». – 2006. - Вып. 93. – С. 99-105.
10. Григорьев А.В. Организация проблемно-ориентированной базы знаний в САПР трубопроводов / А.В. Григорьев // Международная научно-методическая конференция "Автоматизация проектирования и производства в машиностроении". – К.: ИСИ ОУ, 1995. - С.49-50.
11. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции: в 2 т. Синтаксический анализ / Ахо А., Ульман Дж.; пер. Агафонова В.Н.; под. ред. Курочкина В.М. - М.: Мир, 1978. - Т.1. - 614 с.
12. Витяев Е.Е. Извлечение знаний из данных. Компьютерное познание. Модели когнитивных процессов: моногр / Е.Е. Витяев. – Новосибирск: Новосиб. гос. ун-т, 2006. – 293 с.
13. Алексеев С.С. Методы машинного обучения в задачах извлечения информации из текстов по эталону / С.С. Алексеев, В.В. Морозов, К.В. Симаков // Труды 11-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» - RCDL'2009, Петрозаводск, Россия. - 2009.

Надійшла до редакції:
26.02.2011

Рекомендовано до друку:
д-р техн. наук, проф. Скобцов Ю.О.

Abstract

Grigoriev A.V., Morozova O.V. Analysis of the effectiveness and prospects for development of methods for constructing bilateral translators in the task of making intelligent add-on problem-oriented CAD. The paper discusses the effectiveness of the method of constructing two-way translators in the problem of building intelligent add-on problem-oriented CAD tool in the shell for building intelligent CAD systems, as well as ways to improve the process. The problem is solved in the specifics, including the physical semantics of the language level of abstraction, language, skill level expert in the subject area. Discusses a number of implementations of these interfaces for various CAD systems.

Key words: physical semantics of the domain, problem-independent specification language, the language problem-oriented CAD tool shell for building intelligent CAD, Intelligent CAD add-on, methods of construction of double translations.

Анотація

Григор'єв О.В., Морозова О.В. Аналіз ефективності та перспектив розвитку методів побудови двосторонніх трансляторів в задачі створення інтелектуальних надбудов над проблемно-орієнтованими САПР. У роботі обговорюється ефективність методу побудови двосторонніх трансляторів в задачі побудови інтелектуальних надбудов над проблемно-орієнтованими САПР в рамках інструментальної оболонки для побудови інтелектуальних САПР, а так само шляхи підвищення ефективності даного процесу. Завдання вирішується в умовах специфіки, включаючи фізичну семантику мов, рівень абстракції мов, рівень кваліфікації експерта в предметній області. Обговорюється ряд реалізацій таких інтерфейсів для різних САПР.

Ключові слова: фізична семантика предметної області, проблемно-незалежна мова специфікацій, мова проблемно-орієнтованого САПР, інструментальна оболонка для побудови інтелектуальних САПР, інтелектуальна надбудова на САПР, методи побудови подвійних трансляторів.

© Григор'єв А.В., Морозова О.В., 2011