

УДК 004.896

А.В. Григорьев, О.В. Морозова
Донецкий национальный технический университет
grigorie@r5.dgtu.donetsk.ua, olmalyavka@gmail.com

Анализ существующих способов создания интерфейса «языки формальных спецификаций — проблемно-ориентированные языки»

В работе выполнен анализ существующих методов организации интерфейсов «языки формальных спецификаций — проблемно-ориентированные языки». Определены достоинства и недостатки существующих методов с точки зрения задачи организации таких интерфейсов в инструментальной оболочке для построения интеллектуальных САПР. Предложена общая структура построения такого интерфейса.

Проблемно-независимый язык, язык формальных спецификаций, язык проблемно-ориентированной САПР, CASE-технологии, база знаний

Введение

Главный путь повышения эффективности САПР в настоящее время – это повышение уровня их интеллектуальности, включая использование баз знаний, нейронных сетей и т.д. Если существующий САПР не в полном объеме реализует существующую методiku проектирования устройств в данной предметной области, то можно повысить уровень его эффективности, надстраивая над ним базу знаний (БЗ) для синтеза (редактирования, моделирования, документирования) решений по требуемой методике проектирования [1].

Данную методiku проектирования можно записывать на проблемно-независимом языке на языке формальных спецификаций (ЯФС).

В настоящее время имеется ряд технологий, основой которых являются языки формальных спецификаций. Одной из самых популярной технологий является CASE-технология. Суть работы CASE-технологии - это создание программы в модельном виде (символьном или графическом) на проблемно-независимом языке с переводом его в проблемно-ориентированный язык (язык предметной области).

Актуальной задачей при современных тенденциях развития средств автоматизированного проектирования является, построение прямых и обратных систем перевода. Методы перевода из ЯФС в язык предметной области решают задачи: автоматизации создания баз знаний проектирования для любой предметной области в проблемно-независимой форме путем обучения по прототипам; обеспечения передачи готовых решений в полученной проблемно-независимой форме

представления на тот или иной язык проблемно-ориентированного САПР.

Отдельно рассматривается уровень абстракций языка проблемно-ориентированного САПР, т.е. отнесение его к одному из возможных этапов проектирования [1]:

– структурного; функционально-логического;

– уровня количественной макромодели;

– уровня количественной микромоделли.

Для того чтобы построить эффективные методы решения данной задачи необходимо сделать обзор возможностей, а так же достоинств и недостатков уже существующих методов.

Целью данной работы является:

– обзор существующих методов организации интерфейсов ЯФС и проблемно-ориентированных языков с выделением их достоинств и недостатков с точки зрения общей задачи;

– выявление актуальных задач в проблеме построения мета-эвристической оболочки (МЭО) для построения интеллектуальных САПР в режиме умного эксперта;

– построение общей структуры двухстороннего интерфейса.

Уровень квалификации эксперта в базовом методе

Важной тенденцией современных интеллектуальных оболочек (ИО) для создания экспертных систем (ЭС) является исключение из процесса создания ЭС инженера по знаниям в пользу эксперта в предметной области, в данном случае – проектировщика – пользователя САПР. При этом важно, что бы:

– инструментальные средства ИО по семантике и по форме построения были доступны эксперту в предметной области;

– «готовые» интеллектуальные компоненты были ориентированы на уровень подготовки типичного проектировщика.

Предполагается ориентация ИО на уровень квалификации эксперта в предметной области по такой градации [1]:

– «умный» эксперт, способный явно задать методику проектирования в форме продукционной БЗ;

– «средний» эксперт, способный задать основные составляющие методики проектирования, что позволяет ИО построить соответствующую БЗ в полуавтоматическом режиме;

– «глупый» эксперт, не способный помочь ИО в задаче построения БЗ, но могущий предоставить «достоверные» решения, что вынуждает ИО строить соответствующую БЗ в автоматическом режиме.

В базовой теории мета-эвристических оболочек в основном решены задачи связанные с организацией работы в режиме умного эксперта.

Однако имеет место задача развития методов решения данных задач связанные с повышением их эффективности и универсальности.

Перед «умным» экспертом ставится ряд задач:

- создание единичного модуля знаний;
- создание системы модулей знаний;
- организация вывода на системе модулей знаний;
- организация вывода на одном модуле знаний;
- организация интерфейса с проблемно-ориентированном САПР;
- доизобретение.

Одной из самых важных задач является задача построение интерфейса с проблемно-ориентированном САПР.

Рассмотрим базовые инструментарий, имеющийся в мета-эвристической оболочке, для решения данной задачи и предложим методы повышения их эффективности.

Модель предлагаемого интерфейса может быть представлена следующим образом [1]:

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

где:

Go - грамматики языков представления моделей в МЭО;

Fo - формат внутренних структур данных представления моделей в МЭО;

Mo - описания прототипов в данной проблемной области на языках МЭО (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.

Общая характеристика CASE-технологии и ЯФС

CASE-технология представляет собой методологию проектирования информационных систем, а также набор инструментальных средств, позволяющих в наглядной форме моделировать предметную область, анализировать эту модель на всех этапах разработки и сопровождения информационных систем и разрабатывать приложения в соответствии с информационными потребностями пользователей [2].

CASE-средства можно классифицировать по следующим признакам:

- применяемым методологиям и моделям систем и БД;
- степени интегрированности с СУБД;
- доступным платформам.

Большинство существующих CASE-средств основано на методологиях структурного или объектно-ориентированного анализа и проектирования, использующих спецификации в виде диаграмм или текстов для описания внешних требований, связей между моделями системы, динамики поведения системы и архитектуры программных средств [3].

Язык спецификаций — формальный язык, предназначенный для декларативного описания структуры, связей, свойств данных и способов их преобразований, без явного упоминания порядка выполняемых действий и использования конкретных значений данных [4]. Их можно классифицировать по таким категориям:

– универсальные языки с общематематической основой (например, RAISE, Z, API, VDM и др.);

– языки спецификации проблемных областей (например, язык программирования, языки спецификаций предметной области или доменов - DSL и др.);

– специализированные языки спецификации (например, языки таблиц, логики, равенств и подстановок и др.);

– языки, ориентированные на спецификацию параллельных процессов (например, CIP-L, Ada-68, Concurrent Pascal и др.)

Язык формальных спецификаций традиционно рассматривается как средство проектирования. Новое развитие ЯФС получил при рассмотрении и анализе существующего программного обеспечения [4].

Основным направлением стал объектно-ориентированный анализ, который ориентируется на объектно-ориентированное проектирование. Эти направления имеют схожие средства, которые отображают поведения систем, архитектуру построения систем и т.д.

Наиболее популярным средством объектно-ориентированного проектирования является графический язык UML (Unified Modelling Language). Как и подобные ему языки он является хорошим и наглядным средством проектирования, но не способным для проверки правильности.

Еще одним направлением применения ЯФС является использование их как источников для генерации текстов.

Спецификой внутренних проблемно-ориентированных языков является описание модели объекта проектирования. Соответствующие средства построения, преобразования, корректировки модели объекта входят в инструментарий проектировщика.

При создании автоматизированных САПР операции инструментария переходят из разряда ручных операций в разряд автоматизированных, но применяемых к тому же самому объекту проектирования, представленному на том же самом языке проблемно-ориентированного САПР.

Типичные системы, использующие CASE-технологии

В настоящее время имеются разнообразные методы построения подобных трансляторов и CASE-средств. Среди них можно назвать:

– Rational Rose среда для автоматизации этапов анализа и проектирования;

– Vantage Team Builder - интегрированный программный продукт, ориентированный на реализацию каскадной модели жизненных циклов ПО;

– BPwin (AllFusion Process Modeler) - средство функционального моделирования, реализующее методологию IDEF0;

– SoDA продукт предназначен для автоматизации разработки проектной документации на всех фазах жизненных циклов ПО.

Рассмотрим наиболее распространённые CASE-средств Rational Rose и BPwin (AllFusion Process Modeler).

Rational Rose использует диаграммы типа UML для построения программ на разных языках: СИ, Ява и др. Имеется также подход для построения обратных трансляторов, т.е. анализ соответствующих языков программирования с выявлением в них элементов диаграмм UML. Основой такого подхода является входной язык - язык формальных спецификаций, представленный в виде UML диаграмм, который несет в себе смысл проблемно-независимого языка, на котором можно построить и решить поставленную задачу. UML диаграммы несут смысл объектно-ориентированного подхода, т.е. выявление объектов, классов, методов классов и др., что обеспечивает связь между разными объектами [5].

Важными качествами реализованного подхода в работе Rational Rose является универсальность и достаточная гибкость.

Полная совокупность диаграмм позволяет однозначно и полно построить программу на некотором языке программирования. Т.е. из задачи, решенной на ЯФС и записанной в виде полного набора UML диаграмм, получается решение задачи на языке программирования.

Спецификой данного подхода является разработка диаграмм разного вида, которые могут адаптироваться на требуемую предметную область, т.е. на проблемно-ориентированный язык.

Пример диаграммы UML классов в среде Rational Rose, который состоит из 2 классов, представлен на рис.1. Полученный код представлен на рис.2.

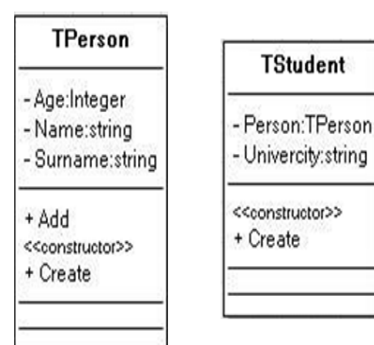


Рисунок 1 - Диаграмма классов

BPwin (AllFusion Process Modeler) - средство функционального моделирования, реализующее методологию IDEF0-IDEF3. Методология IDEF0, представляет собой совокупность методов, правил и процедур, предназначенных для построения

функціональної моделі об'єкта якої-либо предметної області. Функціональна модель IDEFO отображает функціональну структуру об'єкта, т.е. производимые им действия и связи между этими действиями. Методология IDEF может использоваться для моделирования широкого круга систем и определения требований и функций, а затем для разработки системы, которая удовлетворяет этим требованиям и реализует эти функции.

```

type
  TPerson = class
  public
    constructor Create;
    procedure Add(Name : String;Surname :
String;Age : Integer);

  strict private var
    Name:String;
    Surname:String;
    Age:Integer;
  end;

  TStudent = class
  strict private var
    Person:TPerson;
    Univercity:String;

  public
    constructor Create;
  end;

{$R *.res}
constructor TPerson.Create;
begin
  inherited Create;
end;

procedure TPerson.Add(Name :
String;Surname : String;Age : Integer);
begin
end;

constructor TStudent.Create;
begin
  inherited Create;
end;

```

Рисунок 2 - Сгенерированный код по диаграмме классов

Для уже существующих систем IDEF может быть использованы для анализа функций, выполняемых системой, а также для указания механизмов, посредством которых они осуществляются.

Предлагаемый подход к формированию языка формальных спецификаций соответствует задаче построения диаграммы типа диаграмм BPWin (AllFusion Process Modeler), но специализированные на данную предметную область, а именно - построение диаграмм, отражающих физическую семантику предметной области. В этом случае задача отражения такого рода диаграммы в проблемно-ориентированный язык САПР предполагает выявление в данном языке средств построения блоков, связей, потенциалов, массивов и т.д.

В рамках базовой модели интерфейса в МЭО ранее был разработан свой интерфейс для

работы с ЯФС и языком проблемно-ориентированного САПР [6], где на входе подаётся разработанный модуль знаний (рис. 3) и на выходе получается готовый объект на языке проблемно-ориентированного САПР ЛИСП (рис.4).

```

Труба.прямая.Мас_верх[1]:T1 = «10,10,0», T2
=«20,10,0», диаметр =«10», материал = «сталь»
Труба.прямая.Мас_верх[1] : T2-
Труба.прямая.Мас_верх[2]: T1,соединение =
«сварка»
Труба.прямая.Мас_верх[2] : T1 = «20,10,0»,
T2 =«35,10,0», диаметр =«10», материал =
«сталь»
Труба.прямая.Мас_верх[1] : T2-
Труба.уголок.Мас[1]: T1,соединение =
«сварка»
Труба.уголок.Мас[1] : T1 = «35,10,0», T2
=«40,15,0», диаметр =«10», материал =
«сталь»
Труба.уголок.Мас_верх[1] : T2-
Труба.прямая.Мас_прав[3]: T1,соединение =
«сварка»
Труба.прямая.Мас_прав[3] : T1 = «40,15,0»,
T2 =«40,20,0», диаметр =«10», материал =
«сталь»
Труба.прямая.Мас_прав [3] : T2-
Труба.уголок.Мас[2]: T1,соединение =
«сварка»
Труба.уголок.Мас[2] : T1 = «40,20,0», T2
=«35,25,0», диаметр =«10», материал =
«сталь»
Труба.уголок.Мас[2] : T2- Труба.прямая.Мас_
низ [4]: T1,соединение = «сварка»
Труба.прямая.Мас_низ[4] : T1 = «35,25,0», T2
=«20,25,0», диаметр =«10», материал =
«сталь»
Труба.прямая.Мас_низ[4] : T1-
Труба.прямая.Мас_низ [5]: T1,соединение =
«сварка»
Труба.прямая.Мас_низ[5] : T1 = «20,25,0», T2
=«10,25,0», диаметр =«10», материал =
«сталь»

```

Рисунок 3 - Входной модуль знаний

```

1
POINT--->
8
(1 (10 10) 0 10)
(3 (20 10) 0 10)
(5 (35 10) 0 10)
(7 (40 15) 0 10)
(9 (40 20) 0 10)
(11 (35 25) 0 10)
(13 (20 25) 0 10)
(14 (10 25) 0 10)
TRUB'S--->
7
(1 0 1 1 (1 1 (nil)) (2 3 (1 2)) (0 T
T))
(2 0 1 2 (1 3 (1 1)) (2 5 (1 3)) (0 T
T))
(3 3 4 1 (1 5 (1 2)) (2 7 (1 4)) (0 T
T))

```

Рисунок 4 - Выходной ЛИСП-файл

Предлагаемая структура двойного интерфейса

Исходя из опыта разработки интерфейса, стали очевидны недостатки базового подхода. Нужны общие подходы для решения задачи в целом. Перечислим специфику общих подходов задачи [1]:

- физическая семантика проблемно-ориентированных языков и языков спецификаций соответствующего уровня абстракции;

- формы, доступные для передачи или получения данных о структуре объекта проектирования в проблемно-ориентированном САПР: грамматика проблемно-ориентированных языков и формат внутренних структур данных;

- формы, доступные для передачи или получения данных о структуре объекта проектирования в интеллектуальных оболочек: грамматика языков спецификаций и формат внутренних структур данных.

Исходя из специфики языка формальных спецификаций и специфики проблемно-ориентированных языков, существующего набора предметных областей, предлагается следующая структура методов построения двойного интерфейса (рис. 5).



Рисунок 5 - Структура двухстороннего интерфейса

Выводы

Современные CASE-инструменты охватывают обширную область поддержки многочисленных технологий проектирования ИС: от простых средств анализа и документирования до полномасштабных средств автоматизации.

При анализе существующих подходов к созданию CASE-средств можно выявить их достоинства и недостатки.

Достоинства этих подходов состоят в разработке различных типов диаграмм, адаптированных на различные предметные области, т.е., например, моделирование бизнес процессов в BPwin (AllFusion Process Modeler), моделирование баз данных в Rational Rose.

Недостатками рассмотренных систем является многочисленность диаграмм и необходимость подготовки пользователя в области программной инженерии. Чтобы реализовать этот подход необходимо привлечь программного инженера. Он будет создавать методику проектирования (диаграммы), при общении с экспертом в предметной области. С помощью программного инженера решается проблема связи предметной области и ЯФС.

Перспективой дальнейшего развития является создание нового подхода преобразования ЯФС в язык предметной области, в котором происходит удаление инженера по знаниям. Данный подход будет опираться на создание проблемно-независимого языка формальных спецификаций, семантика и синтаксис которого интуитивно понятен эксперту предметной области. Разработанный язык будет использовать такие понятия как блок, тип блока, данное тип данного, массивы блоков и данных, структурные связи между блоками, массивы связей и значение свойств блоков.

Список литературы

1. Григорьев А.В. Построение двухсторонних трансляторов в задаче создания интеллектуальных надстроек над проблемно-ориентированными САПР / А.В. Григорьев, О.В. Морозова // Сборник трудов XI международной научной конференции им. Т.А. Таран. – К.: Просвита, 2011. – С. 68-75.
2. Вендров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем / А.М. Вендров. – М.: Финансы и статистика, 1998. – С. 176.

3. Model-driven Generative Development of Measurement Software / Martin Monperrus, Jean-Marc Jezequel, Benoit Baudryz, Joel Champeaux, Brigitte Hoeltzenerx // Author manuscript, published in "Software and Systems Modeling (SoSyM) tba (2010), inria-00504670, version 1 - 21 Jul 2010. – P. 1-22.

4. Петрухин В.А. Методы и средства инженерии программного обеспечения / В.А. Петрухин, Е.М. Лаврищева [Электронная лекция] – Режим доступа: <http://www.intuit.ru/department/se/swebok/6/>.

5. Kresimir Fertalj. A Source Code Generator Based on UML Specification / Kresimir Fertalj, Mario Brcic // International journal of computers and communications. – 2008. – Vol.2. - P.10-19.

6. Малявка О.В. Анализ интеллектуальной надстройки над проблемно-ориентированной САПР для задачи реконструкции / О.В. Малявка, А.В. Григорьев. – Материалы V юбилейной научно-технической конференции молодых ученых та студентов [Электронная публикация]. — Донецьк, ДонНТУ — 2009. – С. 99-102.

Надійшла до редколегії 21.10.2011

О.В. ГРИГОР'ЄВ, О.В. МОРОЗОВА

Донецький національний технічний університет

A.V. GRIGORIEV, O.V. MOROZOVA

Donetsk national technical university

**Аналіз існуючих способів створення інтерфейсу
«мови формальних специфікацій - проблемно-орієнтовані мови»**

У роботі виконано аналіз існуючих методів організації інтерфейсів «мови формальних специфікацій - проблемно-орієнтовані мови». Визначено переваги і недоліки існуючих методів з точки зору завдання організації таких інтерфейсів в інструментальній оболонці для побудови інтелектуальних САПР. Запропоновано загальну структуру побудови такого інтерфейсу.

Проблемно-незалежна мова, мова формальних специфікацій, мова проблемно-орієнтованого САПР, CASE-технології, база знань

Analysis of existing methods of creating an interface "formal specification languages - problem-oriented languages"

In the analysis of existing methods of the interface "formal specification languages - problem-oriented languages." Identified strengths and weaknesses of existing methods in terms of organizing these interfaces in the shell tool for building intelligent CAD systems. A general framework for constructing such an interface.

Problem-independent language, formal specification language, the language problem-oriented CAD, CASE-technology knowledge base