

ИНТЕРФЕЙС ТАБЛИЧНОГО ПРОЦЕССОРА EXCEL И СПЕЦИАЛИЗИРОВАННОЙ ОБОЛОЧКИ ДЛЯ СИНТЕЗА ИНТЕЛЛЕКТУАЛЬНЫХ САПР

Григорьев А.В., Бондаренко А.В., Шойхеденко А.В.

Кафедра ПМИИ, ДонГТУ

grigorie@pmi.donetsk.ua

Abstract

Grigoriev A.V. The interface of the tabulated processor EXCEL and specialized environment for synthesis intelligent CAD. Two problems are decided: 1) problem of organization of the interface of tool intelligent environments with problem-oriented CAD; 2) problems of creation dynamic of computing models.

1. Постановка проблемы.

Структура [7] специализированной оболочки для построения интеллектуальных САПР и АСНИ - мета-эвристической оболочки (МЭО) включает в себя средства для организации интерфейсов с проблемно-ориентированными САПР. Интерфейс [6] есть совокупность технических и/или программных средств, обеспечивающих сопряжение двух и более элементов системы для их совместного функционирования в этой системе. Целью создания указанных интерфейсов является обеспечение ряда новых возможностей, возникающих при совместном функционировании МЭО и проблемно-ориентированных САПР, таких как:

- 1) создание в среде оболочки интеллектуальных средств автоматизации построения моделей для проблемно-ориентированных САПР;
- 2) использование систем моделирования проблемно-ориентированных САПР, для решения задач исследования моделей, построенных в МЭО;
- 3) обучение синтезированных интеллектуальных САПР на множестве примеров, накопленных в прочих САПР;
- 4) преобразование структуры моделей объектов для решения новых задач в среде проблемно-ориентированных САПР и т.д.

В настоящее время задаче построения подобных интерфейсов уделяется недостаточно внимания. Все существующие интеллектуальные системы проектирования могут быть отнесены в основном к двум классам. К первому классу относятся интеллектуальные оболочки, предназначенные для создания экспертных систем в различных предметных областях и обладающие мощными средствами работы со знаниями, но не имеющие полного набора форм представления, средств исследования и документирования моделей, необходимого для создания полноценного САПР в любой предметной области (KADS, CAKE, ЭЛЬФ, GURU, BESS и т.д.). Ко второму классу относятся сложные проблемно-ориентированные или универсальные САПР, имеющие широкие наборы форм представления, мощные средства моделирования и документирования решений и слабые собственные средства для работы с базами знаний (AutoCAD-13, PCAD, СПРУТ и т.д.).

Разрешение этого противоречия возможно за счет создания интерфейсов проблемно-ориентированных САПР с интеллектуальными экспертными оболочками. Предлагаемая интеллектуальная экспертная оболочка МЭО снабжена комплексом средств и методов для организации такого рода интерфейсов. В настоящее время на этой основе реализованы такие интерфейсы МЭО: с интеллектуальным САПР реконструкция паро-газовых установок теплоэлектростанций СПРУТ; с САПР решения экономических задач - табличный процессор EXCEL.

Выбор EXCEL для создания интерфейса был определен следующими соображениями. К наиболее актуальным предметным областям в данное время можно отнести экономику. Среди систем, предназначенных для создания и исследования экономических проектов, в том числе и интеллектуальных, особую роль занимают табличные процессоры. Программа калькуляции электронных таблиц и деловой графики Excel [4,5] является последним звеном в цепи развития табличных процессоров. Как представитель класса САПР EXCEL обладает мощными средствами моделирования и документирования полученных решений. В силу своей простоты и удобства он стал массовым средством для решения разнообразных экономических задач. Вместе с тем, по причине неразвитости собственных средств работы со знаниями, EXCEL имеет и ряд недостатков. Так, при изменении состава входных и выходных данных в той же задаче пользователь должен либо решать задачу подбора на существующем решении, либо составлять на Excel обратную модель. Создание интерфейса "МЭО-EXCEL" может обеспечить автоматизацию процесса изменения структур ("реконструкции") табличных моделей EXCEL. Кроме того, существует необходимость создания базы знаний о методах решения экономических задач различных классов в среде Excel. Целесообразно создать интерфейс EXCEL с интеллектуальной мета-эвристической оболочкой для использования Excel в качестве математического аппарата моделирования (проектирования) экономических проектов, их документирования, а так же в качестве источника прототипов решений для создания базы знаний путем обучения по примерам.

2. Методы организации интерфейса

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 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.

Охарактеризуем формы представления моделей в МЭО:

1. Грамматика языка представления моделей в МЭО (Go) описана в [10].

2. Модель структур [8] 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 [1], описание которых совмещено с описанием структур (So) и представлено в формате DBF.

Особенности форм представления моделей в EXCEL:

1. Внешний язык описания EXCEL-моделей (Gs) может рассматриваться как язык описания простых вычислительных моделей, интерпретируемых по потоковому алгоритму (см. ниже);

2. Имеется ряд внутренних форматов данных, предназначенных для экспорта-импорта программ в другие программные системы, включая формат SDI. По существу формат SDI представляет собой простой и наглядный язык текстовых описаний EXCEL-модели ($\in Gs$).

Тексты, как форма описания прототипов в объектно - ориентированном САПР, предполагают выполнение таких подпроцедур создания интерфейса:

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);
3. Создание транслятора $Go \leftrightarrow Gs(So)$.

С учетом вышесказанного и был реализован интерфейс "внутренний формат описания моделей EXCEL" - "внутренний формат представления моделей МЭО". Для построения интерфейса была определена грамматика описания исходных текстов в формате SDI - фактически это Gs, проведен семантический анализ грамматики Gs на выявление в ней элементов концептуальной модели, представленной форматами данных So, и создан управляемый компилятор $Gs \leftrightarrow So$.

Остановимся вкратце на главных особенностях семантического анализа описаний моделей в грамматике Gs с целью выявления в них элементов концептуальной модели. Семантический анализ предполагает такие этапы:

1. Выявление точек - единичных структурных составляющих модели [8];
2. Определение свойств точек.
3. Последовательная агрегация выявленных структурных составляющих в структурные единицы различных уровней иерархии представления модели, в соответствии с характером общности их свойств, исходя из особенностей представленной концептуальной модели, включая, в частности: экземпляры типов блоков, свойств, связей, базовых функций; массивы экземпляров типов; типы; библиотеки.

С формальной точки зрения в этом случае последовательно решались две задачи:

1. Задача каузации [6], состоящая в установлении связи двух явлений или фактов, влияния одних явлений или фактов на другие.
2. Задача кластеризации [6], задающей способ разбиения объектов или явлений на классы на основании некоторого отношения близости в пространстве признаков.

Особенности постановки и методов решения данных задач в нашем случае определяются особенностями используемой концептуальной модели предметной области. Решение данных задач предполагает создание каузальных сетей или сценариев. Рассмотрим особенности построенных сценариев диалога. В общем случае, агрегация точек предполагает выявление возможных неоднозначностей, неточностей, что требует уточнение у пользователя в ходе диалога типов блоков и связей, характерных точек и прочего. Разработан единый обобщенный сценарий диалога процесса работы с экспертом, снабженный целым рядом дополнительных приемов и средств для расширения, уточнения и проверки описаний прототипов на непротиворечивость и полноту. На каждом этапе система выполняет доопределение и уточнение модели объекта посредством использования аппарата динамических недоопределенных вычислительных моделей [1]. Доопределение модели выполняется под контролем пользователя. В дальнейшем выполняется формирование описания модели во внутренней форме представления, принятой в оболочке (DBF), построенной в соответствии с заданным описанием концептуальной модели.

3. Задача преобразования EXCEL-моделей в новые формы

Наиболее сложной проблемой в рассмотренной задаче создания интерфейса являлось преобразование EXCEL-модели из исходного вида, связанного с изначальной постановкой задачи, в вид, соответствующий новой, необходимой постановке задачи. Авторами предложен подход к решению этой задачи, основанный на использовании нового метода построения полностью динамических недоопределенных вычислительных моделей. Суть концепции недоопределенных вычислительных моделей, предложенная А.С. Нарипьяни [1] в развитие работ Э.Х. Тыгу [2], заключается в том, что значениям переменных X приписываются недоопределенные значения, связанные с областью возможных значений (перечисление, интервал). Значения связываются друг с другом ограничениями, представленными в виде некоторых математических выражений $r \in R$. Интерпретация этих ограничений по потоковому алгоритму, позволяет уточнять связываемые ими

недоопределенные значения. Представления EXCEL - программы как простой вычислительной модели возможно следующим образом. Множество заполненных ячеек таблицы может рассматриваться как два множества:

1. Ячейки, содержащие константы различных типов. Их можно рассматривать как входы EXCEL-модели.

2. Ячейки, содержащие математические выражения - формулы. Результаты расчета по формулам могут являться и входами для прочих формул. Такие ячейки могут рассматриваться как "условные" выходы EXCEL-модели. Возможны формулы, результаты расчета которых нигде более не используются и которые могут рассматриваться как "безусловные" выходы EXCEL-модели.

Ограничения, накладываемые на значения переменных, имеют такой вид:

1. Для констант они имеют вид равенств некоторой переменной заданного типа конкретному значению.

2. Для переменных, рассматриваемых как результат работы формул, они имеют вид универсума значений U , возможного для данного типа функций (логические, арифметические и т.д.). В этом случае формулы являются формой представления единичных отношений $r \in R$.

Задание нового значения какой-либо константе или изменение той или иной формулы автоматически вызывает процесс интерпретации модели, который в силу простейшего вида ограничений может выполняться за один проход. Это означает, что нет необходимости в поиске альтернативных форм представления формул, необходимых для расчета любой компоненты формулы через прочие, имеющиеся в ее составе.

Возможны следующие изменения ограничений в данной вычислительной модели:

1. Ряду выходов, т.е. формул может приписываться однозначное значение, т.е. осуществляется перевод этой ячейки из разряда выходов EXCEL-модели в разряд ее входов. Это требует:

- Введения равенств как формы ограничения вместо прежней принадлежности к универсуму значений U ;

- Исключения, исходя из особенностей представления моделей в EXCEL, формулы из ячейки, прежде соответствующей данному выходу, т.е. исключения данного отношения из множества отношений R .

2. Ряду входов (констант) может приписываться стать выходами EXCEL-модели, значение которых неизвестно. Это требует:

- Изменение каждого соответствующего ограничения с равенства на принадлежность к универсуму значений U ;

- Переопределения ряда единичных отношений $r \in R$, куда прежде входил данный вход, для перевода их в формы, обеспечивающие интерпретацию ограничений в сторону выходов, прежде связанных с этим входом;

- Включения, исходя из особенностей представления моделей в EXCEL, формулы в данную ячейку, т.е. нового отношения, которое должно соответствовать данному выходу, в множество отношений R .

Возникает задача построения механизма перестройки структуры недоопределенных вычислительных моделей, т.е. множества отношений R . Данная задача не может быть решена в рамках классического подхода к построению недоопределенных вычислительных моделей [1], предполагающего статически заданное множество отношений R . В настоящее время известно решение данной задачи, полученное в рамках так называемого "объектно-ориентированного" подхода к построению динамических вычислительных моделей [3]. Суть данного подхода состоит в следующем:

1. Вычислительная модель представляет собой ряд локальных статических вычислительных моделей, построенных в форме семантических сетей на основе объектов - фреймов, связанных между собой бинарными отношениями. Статические модели обеспечивают монотонность интерпретации отношений, необходимых для классического подхода в недоопределенным вычислительным моделям.

2. Локальные вычислительные модели динамически связаны между собой посредством ссылок на объекты друг - друга.

3. Аппарат ссылок позволяет удалять и добавлять локальные модели, изменяя их ссылки.

Недостатки данного подхода:

1) Методы представления вычислительной модели не соответствуют поставленной задаче;

2) Требуется построить не частично, а полностью перестраиваемые вычислительные модели, при этом отсутствие статических моделей возможно в силу простоты интерпретации рассматриваемого варианта вычислительных моделей.

Необходимо решение задачи в новой постановке.

4. Метод построения динамических вычислительных моделей

4.1. Математическая постановка задачи

Рассмотрим вычислительную модель, описываемую тройкой $C = (X, Y, F)$, где:

$X = \{x_i\}, i = 1..N$ - множество входных переменных;

$Y = \{y_j\}, j = 1..M$ - множество выходных (зависимых) переменных;

$F = \{f_j\}, j = 1..M$ - множество функций, связывающих входные и выходные переменные.

Каждая функция f_j имеет вид:

$y_i = f(X_i, Y_i)$, где $X_i \subseteq \{x_1, \dots, x_N, \emptyset\}$, $Y_i \subseteq \{y_1, \dots, y_M, \emptyset\}$. (*)

Полученная модель $C = (X, Y, F)$ позволяет по известным входным параметрам рассчитывать зависимые переменные за одну итерацию.

В модели (*) X' параметров необходимо сделать выходными путем введения Y' входных параметров, при том, что $X' \subseteq X$ и $Y' \subseteq Y$. Фактически требуется построить новые функции f'_i , которые бы вычисляли новые выходные параметры по оставшимся старым и введенным новым входным. Т.е., если принять $X'' \subseteq X \setminus X'$ и $Y'' \subseteq Y \setminus Y'$, тогда для $\forall y_i \in Y'' \cup X'$ ищем f'_i , такие что: $y_i = f'_i(X'', Y'_i)$, при $X'_i \subseteq X'' \cup Y' \cup \emptyset$, $Y'_i \subseteq Y'' \cup X' \cup \emptyset$. Старые функции, ранее соответствующие Y' , исключаются (**)

4.2. Метод решения для частного случая.

Введем ряд определений.

Функция f не существенно зависит от параметра x_i , если любое изменение параметра x_i не изменяет значения функции f .

Функция f существенно зависит от параметра x_i , если какое-либо изменение параметра x_i изменяет значение функции f .

Алгоритм 1.

Дано:

$C = (X, Y, F)$ - исходная вычислительная модель и пара (x_k, y_j) , при этом:

$y_j \in Y'$ - новый входной параметр;

$x_k \in X'$ - новый выходной параметр.

Получить: $C' = (X_n, Y_n, F')$, - новая вычислительная модель.

где: $X_n = (X \setminus x_k) \cup y_j$, $Y_n = (Y \setminus y_j) \cup x_k$, $F' = \{f'_i\}$ при $i = 1, \dots, M$.

Описание алгоритма.

1. Если y_j не существенно зависит от x_k , то решения нет.

2. В наборе первоначальных формул $y_i = f_i(X_i, x_k, Y_i)$ для $\forall i$ вместо всех функций $y_m \in Y_i$, существенно зависящих от x_k , подставим тела их функций f_m . Обозначив подмножество y_m , не существенно зависящих от x_k , как Z_i , получим новую зависимость для всех $i \neq j$: $y_i = g_i(X'', y_j, Z_i, x_k)$, где:

x_k - рассматривается как новый выход для всех y_i при $i \neq j$;

y_j - рассматривается как элемент множества входов.

3. Результат шага 2 для $i = j$ приведем к виду $g_j(X'', Z_j, x_k) - y_j = 0$.

Найдя аналитическое решение данного уравнения (метод решения рассмотрен ниже в пункте 4.4), получим: $y_j = x_k = f_k(X'', y_j, Z_k)$ при $y_j = x_k$.

4.3. Метод решения общей задачи.

Алгоритм 2.

Дано:

$C=(X, Y, F)$ - исходная вычислительная модель;

$X' \in X$ - множество новых результатов (выходных параметров);

$Y' \in Y$ - множество новых исходных данных (входных параметров).

Получить: новая вычислительная модель $C'=(X'' \cup Y', Y'' \cup X', F)$.

Описание алгоритма.

1. Если $X' = \emptyset$, то $C' = C$. Алгоритм успешно завершён. Конец.

2. Определим $V = X' \times Y'$.

3. Пока $V \neq \emptyset$: Берем $(xk, yj) \in V$.

4. Если алгоритм 1 с параметрами (C, xk, yj) успешен (результат сохраняется в C'), то вызвать алгоритм 2 с параметрами $(C', X' \setminus xk, Y' \setminus yj)$. Результат последнего вызова, а также его успешность являются результатами работы алгоритма. Конец.

5. Если вызов алгоритма 1 с параметрами (C, xk, yj) неуспешен, то $V = V \setminus \{(xk, yj)\}$, повторить шаг 3.

6. Если $V = \emptyset$, то алгоритм потерпел неудачу. Конец.

Таким образом, решение общей задачи сводится к частной.

4.4. Метод поиска аналитического решения уравнения.

Задача преобразования модели сводится к задаче нахождения аналитического решения уравнения (см. п. 3 алгоритма 1). Последняя задача по своей сложности является алгоритмически неразрешимой для всего класса функций, имеющих вектор аргументов и значение - скаляр. Однако, большой класс задач характеризуется простыми функциями, для которых найти решение относительно одного из аргументов не сложно, или же для которых явно определены обратные. К такому классу задач относятся, в частности, многие экономические задачи [7]. Решение уравнения проводится рекурсивно в 2 этапа: построение шаблона функции; поиск шаблона в базе знаний. Введем следующий ряд определений.

Выражение - число; переменная из списка допустимых; функтор, за которым в скобках перечислены выражения;

Функция - определяется именем (функтором), списком аргументов и формой, позволяющей вычислять значение функции на основании списка значений, подставляемых в форму на место аргументов функции. Форма представляет собой выражение, в список допустимых переменных которого входят все аргументы функции.

Константная часть - число; переменная из множества "новых" входов - $X'' \cup Y'$; функция, аргументы которой - константные части.

CONST - специальный символ, определяющий произвольную функцию, определенную над входными параметрами, принадлежащими к множеству X'' . Два CONST всегда эквивалентны как элементы шаблона (см. ниже), но равны только в том случае, если равны связанные с ними функции.

Шаблон функции - функция, построенная на основе искомой, все константные части которой заменены символом CONST. Шаблон некоторой функции f будем обозначать $\mathbb{W}(f)$.

Функция f эквивалентна функции g , если $\mathbb{W}(f) = \mathbb{W}(g)$. Обозначение эквивалентности функций по шаблону: $f \in g$.

Алгоритм 3.

Дано: f - функция;

Получить: $\mathbb{W}(f)$ - шаблон функции.

Описание алгоритма.

1. Все константные части заменяются на CONST. Если такие части не найдены, то переход на шаг 3, иначе на шаг 2.

2. Осуществляются алгебраические преобразования, такие как: раскрытие скобок; приведение к общему знаменателю; сортировка аргументов для коммутативных операций в таком порядке, чтобы все аргументы - CONST оказались в конце списка аргументов; приведение подобных; замена производных операций (вычитание, деление) прямыми

(соответственно сложение, умножение) с обращением всех аргументов кроме первого посредством обратных функций (соответственно унарный минус, $1/x$). Если такие операции выполнены, то переход на шаг 1, иначе на шаг 3.

3. Конец.

База шаблонов представляет собой множество кортежей вида: (Шаблон для сравнения - Шс, Результат решения - Рр). Если $\mathcal{W}(f) = \mathcal{W}_s$, то это значит, что решение $f(\dots, x_k)$ относительно x_k эквивалентно Рр. Решение строится следующим образом. На этапе сравнения $\mathcal{W}(f)$ с \mathcal{W}_s всем CONST в \mathcal{W}_s ставятся в соответствие CONST из $\mathcal{W}(f)$. В Рр все CONST заменяются связанными функциями из соответствующих CONST функции f . Полученная форма есть результат решения уравнения $f(x_k) = 0$ относительно x_k . Если ни один шаблон базы знаний не совпадает с построенным шаблоном функции, то построенный шаблон усложняется (например, вместо $x + \text{CONST}$ ставится $\text{CONST} * x + \text{CONST}$), после чего выполняется новый цикл поиска по базе шаблонов. Процесс усложнения идет следующим образом: сначала к переменной без константного множителя добавляется множитель CONST, при этом отмечается, что этот CONST соответствует единице. Если в выражении с операцией сложения нет CONST, в нее тоже добавляется CONST. В ситуации, если поиск во всех случаях неудачен, алгоритм поиска решения терпит неудачу. В дальнейшем пользователь может пополнить базу знаний шаблоном для указанного уравнения. Класс решаемых уравнений зависит от наполненности базы знаний.

Заключение.

Апробация методики создания интерфейсов с указанными проблемно-ориентированными САПР показала, что методика может быть применена и для создания интерфейса с другими проблемно-ориентированными САПР. Предложенная модель динамических вычислительных моделей может с успехом использоваться как при доопределении недоопределенных моделей, созданных интеллектуальной оболочкой САПР, так и для синтеза экономических моделей. Разработанная объектно-ориентированная PASCAL - библиотека аналитических преобразований может найти большую область применения, чем это было изначально задано. Например: аналитическое дифференцирование; вычисления по заранее неизвестным формулам.

Литература

1. Нариньяни А.С. Недоопределенность в системах представления и обработки знаний // Известия АН СССР. Техническая Кибернетика. - 1986. - №5. - С.3-28.
2. Тыгу Э. Х. Концептуальное программирование.
3. Каракозова О.В. Использование динамических вычислительных моделей // Материалы. м. н.-т. конференции КИИ-96 - Инженерия знаний. - С.260-264.
4. Олаф Кох. MS Excel 4.0 ... для пользователя. Москва. Фирма БИНОМ, 1994.
5. Григорьев А. Опыт разработки информационного обеспечения бизнес-планов. Материалы второй международной научно-практической конференции "Регион: стратегия выживания и развития Донбасса" / Донецк: ООО "Лебедь", 1996. - С. 324-326.
6. Толковый словарь по искусственному интеллекту/ Авторы-составители А.Н.Аверкин, М.Г.Гаазе-Рапопорт, Д.А.Поспелов. - М.: Радио и связь, 1992 г.
7. Григорьев А.В., Базалей А.О. Специализированная оболочка для синтеза интеллектуальных САПР и АСНИ (см. в этом же сборнике).
8. Григорьев А.В. Унифицированная концептуальная модель предметной области (см. в этом же сборнике).