

пределов интегрирования, которые являются элементами шифра. Коэффициенты А, В и С можно использовать в качестве элементов ключа рассмотренного шифра.

Выводы

Рассмотренный в статье адаптивный многоалфавитный шифр целесообразно использовать при шифровании объемных текстов. Криптоанализ данного шифра осложнен тем, что элементы криптограммы представляют собой вещественные числа, распределенные по равномерному закону.

Литература

1. Алексеев А.П. Информатика 2007. М.: СОЛОН-ПРЕСС, 2007. – 608 с.
2. Алексеев А.П. Математические методы формирования многоалфавитных шифров замены // ИКТ. Т.7, № 2, 2009. – С. 21-25.
3. Алексеев А.П., Камышенков Г.Е. Использование ЭВМ для математических расчетов. Самара: Парус, 1998. – 190 с.

НОВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 658.512.22

СТРУКТУРА ФОРМАЛЬНОГО ПРЕДСТАВЛЕНИЯ ПРОЦЕССА ПРОЕКТИРОВАНИЯ В ФУНКЦИОНАЛЬНО АДАПТИРОВАННОЙ САПР

Горбачев И.В., Похилько А.Ф.

В статье рассматривается структура функционально адаптированной САПР и принципы работы с ней. Описывается структура формального описания объектов информационного пространства функционально адаптированной САПР и принципы формирования условий.

Введение

Проблема обмена решениями между различными САПР, на сегодняшний день, остается не решенной. Разработанный формат ISO 10303 STEP позволяет передать геометрическую модель объекта из одной системы в другую, но передается только геометрия модели. Логика (дерево) построения модели передать из одной геометрической среды в другую без искажений практически невозможно. Внесение в существующие модели изменений является необходимым элементом инфокоммуникационного взаимодействия в проектной деятельности с использованием современных информационных технологий, так как по существу означает невозможность использования «опыта со стороны» [1].

Сложность проблемы достижения интероперабельности в данном контексте является принципиальной. САД-системы от различных производителей проектирования никогда не будут иметь в точности одинаковый набор операций, так как потеряют свои конкурентные преимущества; так же как невозможно в рыночных условиях заставить всех пользоваться только одной и той же САПР [2].

Предлагается передавать геометрическую модель проектируемого объекта вместе с функционалом, который позволяет ее построить – то есть в виде небольшого программного модуля, содержащего только необходимый для решения текущей задачи набор функциональности – функционально адаптированной САПР (ФА САПР).

За счет включения в ФА САПР конвертора STEP, модель можно передать в любую другую систему. Если возникает необходимость внести в модель изменения, то ФА САПР обладает достаточным функционалом для этого.

Структура функционально адаптированной САПР

В состав ФА САПР входят три модуля:

- исполняемый модуль, который представляет собой графический интерфейс пользователя с процедурами вызова функций графического ядра, областью моделирования (отображения) трехмерной модели проектируемого объекта, и деревом построения объекта, средствами редактирования прочих проектных операций (выбора табличных данных, математических расчетов и др.);

- библиотеки графического ядра: набор перекompilированных библиотек, из состава которых удалена избыточная функциональность. Для построения ФА САПР требуется ядро геометрического моделирования. Для проведения программного эксперимента используется геометрическое ядро Open CASCADE. Исследо-

вание существующих ядер, показало, что для поставленных задач возможно использование графического ядра Open CASCADE, поскольку оно является единственным бесплатным графическим ядром с открытым исходным кодом, предназначенным для построения систем проектирования;

- база данных проекта, которая включает в себя информацию о модели проектируемого объекта и наборе используемой функциональности.

На этапе создания проекта проектирования технического объекта, инженер создает типовую модель объекта. При этом среда построения ФА САПР фиксирует последовательность его действий и используемые функции.

Когда проект готов, инженер вызывает процедуру компиляции, и среда самостоятельно генерирует новый программный модуль и проводит перекомпиляцию библиотек геометрического ядра, оставляя в них лишь требуемую для работы текущего проекта функциональность, и создает базу данных проекта, сохраняя в ней последовательность проектирования. Таким образом, получается функционально адаптированная САПР заданного технического объекта. Далее ФА САПР может быть передана на последующие этапы проектирования и подготовки производства в качестве результата проектирования. С другой стороны, данная ФА САПР может использоваться на рабочем месте проектировщика, занимающегося проектированием одного определенного класса технических объектов.

При различных ситуациях, может возникнуть необходимость расширить либо изменить типовую модель объекта. Если требуемые изменения вписываются в уже имеющуюся функциональность, то они могут быть сохранены в базе данных ФА САПР, как альтернатива в дополнение к имеющемуся проекту. Если же новый набор функциональности выходит за рамки имеющегося проекта, то инженер должен будет вернуться на этап создания проекта, и сгенерировать новую ФА САПР.

Применение такого подхода, на первый взгляд, кажется достаточно сложным. Но это не так. При создании проекта, инженеру не требуется никаких дополнительных навыков и знаний помимо знания своей предметной области. Все остальное среда построения ФА САПР делает сама, инженер же должен создать параметризованную модель объекта, и вызвать процедуру компиляции.

Структура формальной логики процесса проектной операции

Описание процессов возможно с использованием ICOM модели. В нашем случае, описание процесса построения модели, так же возможно с применением методологии ICOM.

Процесс проектирования, описанный в терминах ICOM представлен на рис. 1. На вход работы поступает техническое задание (ТЗ) на проектирование технического объекта (ТО), которое описывается как множество входных параметров вида

$$T3 = \{ a_1; a_2 \dots a_N \}. \quad (1)$$

Управление U на рис. 1 представляет собой набор средств пользовательского интерфейса, который проектировщик использует при проектировании.

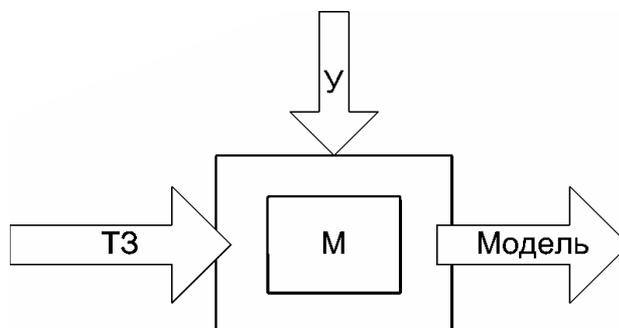


Рис. 1. ICOM модель проекта

Модель ТО – результат проектирования, представленный в виде конкретного решения (конкретной модели), сохраненной в виде обменного файла формата STEP. Так как каждая ФА САПР представлена в виде конкретного проекта, то M на рис. 1 – это механизм, который представляет собой упорядоченную последовательность проектных операций PrO , формирующуюся в процессе проектирования, то есть

$PrO = \{ PrO_1; PrO_2 \dots PrO_N \}$ – множество всех проектных операций.

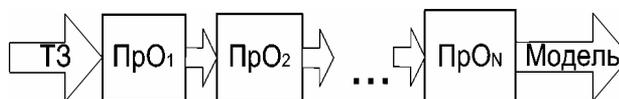


Рис.2. Декомпозиция модели проекта

В простейшем случае, совокупность используемых проектных операций выстраивается в линейную последовательность. Декомпозиция такого процесса представлена на рис. 2. Таким

образом, решением проекта является функция вида

$$F(a_1, a_2, \dots, a_N) \text{ вида} \\ F(a_1, a_2, \dots, a_N) = \text{Пр}O_1 \cup \text{Пр}O_2 \cup \dots \cup \text{Пр}O_N.$$

Данное решение будет верным, так как решение образуется путем фиксации шагов проектирования верного решения.

Вторым важным атрибутом проекта являются проектные параметры (ПрП). Проектные параметры осуществляют логическую связь между проектными операциями. Проектные параметры могут принимать значения, как числовых переменных, так и других типов данных (например, тип материала, нарисованный эскиз, выбранная грань и прочее). В связи с этим, рассмотрим более подробно структуру и механизм работы проектных операций (см. рис. 3).

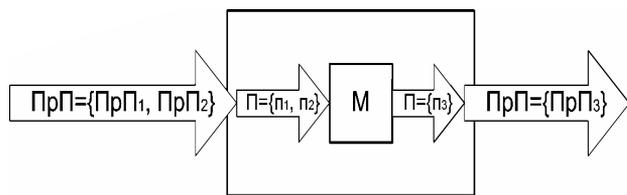


Рис. 3. Структура проектной операции

На вход проектной операции подаются значения проектных параметров $\{\text{Пр}P_1, \text{Пр}P_2\}$, значения которых присваиваются параметрам проектной операции $\{n_1, n_2\}$. Механизм М в данном случае представляет собой последовательность выполненных процедур. Результатом их выполнения является параметр $\{n_3\}$, значение которого присваивается проектному параметру $\{\text{Пр}P_3\}$.

Если механизм исполняемых процедур описывается некой функцией $f(x, y) = z$ и, при выполнении проектной операции, переменной x присваивалось значение $\{n_1\}$, а переменной y – $\{n_2\}$, которым, соответственно, присваивались значения переменных проекта $\{\text{Пр}P_1, \text{Пр}P_2\}$, то после выполнения функции $f(x, y)$ полученное значение z присваивается переменной $\{n_3\}$, а потом переменной проекта $\{\text{Пр}P_3\}$. То есть

$$n_1 = \text{Пр}P_1; \\ n_2 = \text{Пр}P_2; \\ n_3 = f(n_1, n_2), \text{Пр}P_3 = n_3.$$

В общем виде исполняемые процедуры сводятся к выполнению некоторой процедуры $F(X) = Y$, где $X = \{x_1; x_2 \dots x_N\}$ – множество входных параметров, $Y = \{y_1; y_2 \dots y_N\}$ – множество выходных параметров.

Входом и выходом каждой такой проектной операции являются параметры (входные

и выходные соответственно), на определение последовательности выполнения проектных операций они не влияют, так как после выполнения проектной операции значения помещаются в некоторое хранилище параметров, а перед выполнением – из этого хранилища берутся соответствующие значения. Таким образом, гипотетически, может показаться возможной ситуация, при которой для некоторой проектной операции не будет определено значение какого-нибудь из ее входных параметров. Но, при проектировании, инженер не начнет выполнения какого-либо действия, пока у него не будут готовы все данные для его выполнения, а система как раз фиксирует последовательность таких действий. Точнее сказать фиксирует последовательность правильных действий.

Структура формальной модели проекта

Каждый проект в данной системе описывается множеством проектных операций и проектных параметров:

$$\text{Пр} = \{\text{Пр}O, \text{Пр}P\},$$

где $\text{Пр}P = \{\text{Пр}P_1; \text{Пр}P_2 \dots \text{Пр}P_N\}$ – множество проектных параметров;

$\text{Пр}P = \{\text{Пр}O_1; \text{Пр}O_2 \dots \text{Пр}O_N\}$ – множество проектных операций.

Каждая проектная операция описывается множеством операций и множеством параметров:

$$\text{Пр}O = \{O; \text{Вх}P; \text{Вых}P\},$$

где $\text{Вх}P = \{P_1; P_2 \dots P_N\}$ – множество входных проектных параметров;

$\text{Вых}P = \{P_1; P_2 \dots P_N\}$ – множество выходных проектных параметров;

$O = \{O_1; O_2 \dots O_N\}$ – множество всех проектных операций.

Каждая проектная операция представляет собой набор некоторых процедур, которые обрабатываются процессорами определенного типа: текстовым; табличным; математическим; интерактивного ввода данных; геометрическим; обработки ветвлений (условий). В соответствии с типами процессоров $O = (T; Tб; M; B; G; Y)$ – совокупность всех типов операций.

С другой стороны, операции могут состоять из конечного числа операционных процедур $O = (P_1; P_2 \dots P_N)$ определенного типа. Использование процедур разных типов в одной операции не допускается.

Структура формального представления ветвлений

Одним из важных моментов в описании проектной деятельности являются ветвления при создании и выборе альтернатив. Данный тип операций применяется в случаях, когда возникает необходимость создать альтернативные пути решения, например, если необходимо задать выбор из нескольких альтернатив в конструкции технического объекта (в качестве живого примера, можно привести выбор типа хвостовика протяжки). Использование ветвлений (условий) возможно и в некоторых других случаях.

При проектировании технического объекта, проектировщик имеет возможность задать процедуру ветвления, а затем, по отдельности наполнить операциями различные пути в получении решения. Либо создав первое решения, внести на нужном этапе выполнения проекта условие, уже имеющиеся последующие операции станут первым путем ветвления, и проектировщику останется наполнить только другие альтернативы.

Операция ветвления описывается как

$$OB = \{VxП; B; Усл\},$$

где $VxП = \{P_1; P_2 \dots P_N\}$ – множество входных параметров; $B = \{B_1; B_2 \dots B_N\}$ – множество направлений ветвления; $Усл = \{Усл_1; Усл_2 \dots Усл_N\}$ – множество условий ветвления; $Усл_i$ – условие выбора движения по i -му маршруту.

В самом простом случае направление дальнейшего движения выбирается непосредственно пользователем во всплывающем диалогом окне. Допустим, имеется проект, в котором присутствует одна операция ветвления, с двумя маршрутами. Общим решением такого проекта будет функция $F(a_1; a_2 \dots a_N)$ вида

$$F(a_1; a_2 \dots a_N) = ПрО_1 \cup ПрО_2 \cup ((ПрО_3 \cup ПрО_4 \cup \dots \cup ПрО_N) \cap (ПрО_{N+1} \cup ПрО_{N+2} \cup \dots \cup ПрО_{N+M})).$$

При получении конкретного решения функция $F(a_1; a_2 \dots a_N)$ примет один из двух соответствующих видов:

$$F(a_1; a_2 \dots a_N) = ПрО_1 \cup ПрО_2 \cup ПрО_3 \cup ПрО_4 \cup \dots \cup ПрО_N,$$

или

$$F(a_1; a_2 \dots a_N) = ПрО_1 \cup ПрО_2 \cup ПрО_{N+1} \cup ПрО_{N+2} \cup \dots \cup ПрО_{N+M}.$$

Таким образом, сохраняя решение вместе с ветвями решений (условиями), можно сохранить несколько вариантов исполнения технического объекта.

Заключение

Использование предлагаемого подхода в описании информационных объектов ФА САПР позволяет построить интерактивную (диалоговую) среду для построения ФА САПР, решающих задачи проектирования технических объектов на уровне узлов.

Фиксация процесса проектирования, позволяет сохранять последовательность действий проектировщика, для дальнейшего воспроизведения в качестве алгоритма работы автоматизированной системы. Но при линейном процессе проектирования возможно только строгое повторение его действий.

Включение в последовательность операций ветвлений (условий) позволяет строить более гибкие системы, предназначенные для проектирования классов объектов.

В результате такого подхода к построению, становится возможным строить и изменять ФА САПР без привлечения программистов, что для малых и средних предприятий должно стать достаточно привлекательным фактором.

Литература

1. Roberta Charlton CAD interoperability problem still exists // Kubotek USA, http://www.kubotekusa.com/news_events/press_releases/pr_08_01_2005.html
2. Хэмилтон П. Азбука технологий моделирования в MCAD-системах. Часть III. Как технологии MCAD влияют на процесс разработки изделия // CAD/CAM/CAE Observer. №2 (38), 2008. – С. 34-36.
3. Pokhil'ko A.F., Maslyantcin A.A. Toolkit for Support of the Process Approach for an Automated Workplace of the engineer-designer // Interactive Systems: The Problems of Human. Computer Interaction. Ulyanovsk: USTU, 2003. – P. 29-37.
4. Горбачев И.В., Похилько А.Ф. Анализ функциональности модели технического объекта при построении систем проектирования // Труды XII МНПК «Системный анализ в проектировании и управлении». Ч. 2. СПб: Изд. СПбПТУ, 2008. – С. 229-233.
5. Похилько А.Ф., Масляницын А.А., Скворцов А.В., Удовиченко А. В. Формальное представление процесса проектной деятельности в инструментальной инфокоммуникационной среде САПР // ИКТ. Т.6., №1, 2008. – С. 80-83.
6. Похилько А.Ф., Масляницын А.А., Удовиченко А.В., Куприянов А.А. Формализация и анализ процессов проектирования технических объектов // Автоматизация процессов управления. Интегрированные АСУ. №2 (8), 2006. – С.132-137.