Автоматизация проектирования прикладных информационно-вычислительных систем

Ю.А. Флеров, Л.Л. Вышинский

Федеральное государственное учреждение "Федеральный исследовательский центр "Информатика и управление" Российской академии наук", г. Москва, Россия

Аннотация. В статье рассмотрены вопросы автоматизации проектирования и разработки сложных прикладных информационно-вычислительных систем. Кратко изложена технология проектирования прикладных программных систем, основанная на проектном подходе и инструментальном программном комплексе «Генератор проектов». На примере конкретных прикладных систем описаны основные положения такого подхода. В Вычислительном центре РАН в разные годы с помощью «Генератора проектов» был разработан ряд крупных информационных систем. В статье перечислены некоторые из них, дана их характеристика и показана эффективность применения технологии «Генератора проектов».

Ключевые слова: автоматизация проектирования, генератор проектов, генерация программного кода, архитектура клиент-сервер.

DOI 10.14357/20718632180303

Введение

В течение нескольких последних десятилетий кардинально изменились подходы к разработке прикладных информационно - вычислительных систем (ИВС). В 70-х - 80-х годах прошлого века, когда началось широкое использование вычислительной техники, создание ИВС велось относительно небольшими коллективами прикладных математиков и программистов. Постепенно разработка прикладного программного обеспечения (ПО) стала самостоятельной отраслью и в настоящее время представляет собой высокотехнологический, быстро развивающийся сектор мировой экономики. Сейчас на рынке прикладного ПО преобладают программные продукты, которые охватывают значительные области производственной, экономической, управленческой деятельности. Многие системы, завоевавшие рынок, представляют собой большие интегрированные комплексы широкого профиля. Они покрывают значительную область потребностей в информационных технологиях. Однако большие системы требуют соответствующих ресурсов, крупных вложений в закупку, установку, обслуживание и сопровождение. Кроме первичных вложений, неизбежно возникают дополнительные затраты, связанные с настройкой купленных универсальных систем на решение конкретных задач. Поэтому, несмотря на безусловные достоинства высокоразвитых фирменных программных продуктов, часто возникает необходимость создания специализированных систем, ориентированных на автоматизацию конкретных, исторически сложившихся на предприятиях, технологических процессов. Такой подход привлекателен тем, что заказчик может ставить достаточно сложные задачи пе-

ред разработчиком, может напрямую влиять на ход разработки и требовать оперативной реакции на возникшие проблемы. По этим причинам совершенствование технологии разработки прикладных ИВС по-прежнему остается актуальной задачей. Надо сказать, что современные технологии создания прикладных программ существенно усложнились. При этом усложнение прикладных программ связано не столько с усложнением математических моделей, сколько с необходимостью работать с современными информационными технологиями – с базами данных, разнообразным пользовательским интерфейсом, сетевой организацией вычислительных сред и пр. Эти возможности, безусловно, улучшают качество программ, обеспечивая пользователей соответствующей инфраструктурой. Однако, как всякая инфраструктура, она дорого стоит и выливается в многократное увеличение трудоемкости разработки. Поэтому без должного уровня автоматизации процесс создания прикладных систем большой сложности требует больших затрат как на этапах проектирования и написания программного кода, так и в процессе эксплуатации и сопровождения разработанных систем.

В Вычислительном центре РАН несколько лет велись работы по созданию технологии автоматизации проектирования прикладных ИВС. Был разработан инструментальный комплекс "Генератор проектов" (ГП) [1]. Согласно принятому подходу создание новых прикладных ИВС выполняется в три этапа: на первом этапе идет проектирование системы, т.е. создается необходимый комплект документов (проект системы), однозначно описывающий состав, структуру и функции будущей системы. На втором этапе по созданному рабочему проекту генерируется программный код системы. Этот технологический этап в ГП полностью автоматизирован, как и третий, завершающий этап, на котором осуществляется сборка и сдача системы в эксплуатацию. Такая последовательность разработки прикладных ИВС за счет автоматизации второго и третьего этапов позволяет существенно снизить трудоемкость и повысить качество разработки.

Технология ГП постоянно совершенствуется, расширяются ее возможности. В основном, это связано с новыми требованиями, возникающими при разработке новых приложений. Создание с помощью ГП целого ряда весьма успешных проектов в самых различных областях явилось хорошим стимулом развития технологии. В данной статье дано краткое описание основных принципиальных нововведений в ГП и приведен ряд важных практических результатов в разработке средствами ГП прикладных систем.

1. Структура описания проектов прикладных ИВС в технологии ГП

Основной подход к описанию проектов прикладных ИВС в ГП формировался в процессе работы над банковскими системами, которые строились по технологии клиент-сервер:

«КЛИЕНТСКОЕ РАБОЧЕЕ МЕСТО» - «СЕРВЕР» - «БАЗА ДАННЫХ».

В рамках такой архитектуры разрабатываются многие другие многопользовательские программные комплексы с удаленным доступом, с централизованным и децентрализованным хранением информации. Работа таких систем сводиться к выполнению команд и запросов, переданных от пользователей системы – клиентов – к исполнительным компонентам системы – серверам. На клиентском рабочем месте формируются данные запроса, затем они по информационным каналам передаются на сервер, который обрабатывает полученный запрос, обращается при необходимости к базе данных, проводит нужные вычисления, формирует ответ и передает его обратно на клиентское рабочее место. Процедуры формирования и обработки запросов составляют содержательную сторону прикладной задачи, которая должна выполняться системой в рамках данного конкретного сценария. Каждая из таких процедур (например, для банковской системы - просмотр счета клиента банка, выполнение финансовой проводки, выдача выписки или построение дневного баланса) представляет собой отдельную транзакцию системы. Транзакция – это запрос клиента плюс ответ сервера. Перечень и формальное описание всех допустимых в системе транзакций – это фактически спецификации функций системы.

Программная реализация таких ИВС должна вестись в двух направлениях. Одно из них - разработка программных компонент, реализующих всю совокупность содержательных функций системы, ее транзакций. Реализацию этого направления работ должен вести прикладной аналитик, владеющий полным объемом знаний предметной области и способный точно сформулировать механизмы реализации этих функций.

Второе направление - это то, что выше было отнесено к технологии, - программная реализация выполнения этих функций. Здесь осуществляется реализация функций обеспечения нужных протоколов сетевого обмена, программная реализация пользовательского интерфейса, связь с базами данных, обеспечение информационной безопасности и другие важные системные аспекты работы программного комплекса. Это направление работ производится системными программистами. В технологии ГП при разработке конкретных систем такие работы должны выполняться автоматически на основании структуры проекта, описанной прикладным аналитиком или архитектором системы. Одной из основных задач системной организации проекта ИВС является поддержание целостности проекта в части соответствия клиентских и серверных программ обмена, протоколов межпрограммного взаимодействия и тому подобного.

Четкое понимание задач упомянутых двух направлений позволяет отделить работу прикладных аналитиков от работы системных программистов - технологов. Аналитик концентрируется на прикладных задачах, а системный программист на операционных системах, протоколах, языках и пр. Работа аналитика состоит в описании предметной области, всех информационных объектов проекта ИВС, а также всех транзакций, связанных с действиями над этими объектами. Средством формального описания предметной области является тот или иной язык программирования. Основным направлением совершенствования технологии ГП являлось создание и развитие специального базового языка, предназначенного для описания всех компонент проекта разрабатываемых систем. На этом языке могут быть разработаны любые процедуры обработки данных как на клиентском рабочем месте, так и на серверных компонентах. В этом смысле базовый язык является универсальным языком программирования. Он включает в себя традиционные конструкции обычных языков программирования, в частности, языка С. Надо сказать, что в ГП допускается использование фрагментов программ на языке С. Однако с появлением новых возможностей базового языка ГП, использование С в реальных проектах стало весьма ограниченным, т.к. язык ГП оказался достаточно эффективным для реализации большинства задач.

Базовый язык ГП обладает всеми характерными для универсальных языков конструкциями. В нем есть именованные типы данных – строки, числа, перечислимые типы, битовые маски с изменяемыми компонентами, структуры в смысле языка С, массивы. Основные стандартные операторы - это присваивание, условные операторы **if then else**, цикл **for**, цикл **while**, вызов процедуры, выражения. Основной функциональной сущностью языка являются процедуры (функции) с указанием типов параметров. Все очень похоже на обычные стандартные языки.

Кардинальным отличием и преимуществом базового языка ГП является наличие типа данных, допускающего описание широкого класса структурно-параметрических моделей сетевого типа. Структуры данных сетевого типа хорошо известны с 60-х годов, когда были разработаны стандарты под эгидой организации CODASYL. Со временем этот подход был заброшен, хотя он обеспечивал достаточно высокую эффективность при решении сложных прикладных задач. Причиной потери популярности сетевых структур стала относительная сложность манипулирования данными такого типа. Возобладала концепция реляционных структур данных, в рамках которой работают современные СУБД. Возврат к сетевым структурам в ГП оказался возможен в силу высокого уровня автоматизации программирования. Сложности традиционных языков программирования были преодолены благодаря возможностям описания сетевых структур, автоматической генерацией необходимого набора операторов и встраивание их в конструкции базового языка ГП.

Описание сетевых структур в ГП было реализовано в виде специальной конструкции следующего вида:

type <имя типа сетевой структуры> : ***document***
(***record*** <имя записи 1> ***:*** <тип записи 1>***;***
record <имя записи 2> ***:*** <тип записи 2>***; …***
set <имя набора 1**> [owner <**владелец набора 1**>] member <**члены набора 1**>;**
set <имя набора 2**> [owner <**владелец набора 2**>] member <**члены набора 2**>; …**
);

Владельцами и членами наборов являются записи данной структуры. Если владелец набора отсутствует, то этот набор представляет собой простой список. Для каждого типа описанной в проекте сетевой структуры ГП автоматически генерирует весь необходимый в проекте комплект операций, связанных с манипуляциями данными: создание новой сетевой структуры, создание, считывание, модификация, удаление новой записи в структуре, создание нового набора, присоединение, отсоединение записи к набору, поиск записи в наборе по номеру или по ключевым полям, поиск владельца набора, выдача списка членов набора и прочее.

Сетевые структуры в описании проектов являются мощнейшим инструментом описания широкого класса предметных областей. На основе сетевых структур в ГП наряду с реляционными базами данных можно создавать полноценные базы данных сетевого типа.

Основная концепция технологии ГП состоит в том, чтобы сделать системную организацию ИВС независимой от содержательных задач и максимально автоматизировать разработку этой системной части программного кода. Это оказалось возможным только потому, что было принято решение о выборе определенной модели разрабатываемых проектов ИВС. Модель проекта определяется такими сущностями высокого уровня, как база данных, клиентское приложение, типы окон клиентских приложений, формы диалогов, программные серверы, порты серверов, подсистема безопасности и ряд других понятий. Все модельные сущности со своими реквизитами и параметрами должны быть представлены в исходном тексте описания проекта.

Пространство разработки очередного прикладного проекта ограничено отдельным каталогом, в котором размещены текущая версия ГП, исходное описание проекта и директории сгенерированного программного кода. Таким образом, сам ГП, его текущая версия становятся неотъемлемой частью проекта.

Исходное описание проекта - это совокупность текстовых файлов разного типа. Головным, обязательным файлом проекта является файл **<имя проекта>.gen.** Имя файла, оно же имя проекта является его ключевым атрибутом. В головном файле проекта дана структура описания проекта, другими словами, перечень всех проектных документов ИВС.

project *<имя проекта> [«<текстовое название проекта>»]*
{/<опции проекта> }
{ <тип пакета> <имя пакета> }

Имя проекта, которое следует за ключевым словом ***project,*** должно совпадать с именем файла. В опциях проекта задаются его версия, дата последней версии, исполнители проекта, указатели режима генерации и сборки проекта, другая полезная информация. Далее идет перечень компонент описания проекта, которые в технологии ГП называются пакетами.

Пакет - одно из основных понятий генераторного базового языка и представляет собой логически завершенный фрагмент программного текста. Пакет специфицируется типом и имеет уникальное в пределах проекта имя. Имена пакетов важные атрибуты. На них могут быть ссылки в разных местах описания. Каждый пакет представлен в виде отдельного файла в каталоге описания проекта. Имя файла-пакета имеет следующий вид **<имя пакета>.<тип файла>**. Тип файла представляет собой сокращение от наименования типа пакета.

Разные типы пакетов соответствуют различным проектным сущностям. Ниже перечислены основные типы прикладных пакетов, которые составляют описание архитектуры проекта:

package *(*.pkg)* - пакеты описания структур данных и процедур проекта,

database *(*.dbs)* - пакеты описания логических схем баз данных (БД) реляционного типа,

genbd *(*.gbd)* - пакеты описания логических схем БД сетевого типа,

port *(*.prt)*- пакеты описания точек входа в сетевой канал и протоколов обмена информацией через этот порт,

server *(*.srv)* - пакеты описания программных серверов и их функций,

application*(*.app)* - пакеты описания модулей клиентских рабочих мест и их функций,

dlg *(*.dlg)* - пакеты описания пользовательских диалогов ввода информации,

window *(*.wnd),* ***wintable*** *(*.wtb)* - пакеты описания оконного представления информации на клиентских рабочих местах.

В составе генератора, кроме пакетов, перечисленных в головном файле, есть большой набор встроенных системных пакетов разного типа, которые могут быть использованы явно или неявно через модельные сущности в описании проекта.

Совокупность прикладных и системных пакетов образует единое пространство описания проекта. Функции прикладного аналитика (аналитиков), проектирующего систему, состоят в написании всех прикладных пакетов, заявленных в головном файле проекта, и согласование типов формальных и фактических параметров. В ГП осуществляется строгий контроль типов данных и вообще отслеживается целостность, полнота, непротиворечивость и избыточность описания проекта.

При описании проекта решаются три главные задачи: во-первых, разрабатывается архитектура системы, ее структурная модель, во-вторых, разрабатывается структура данных системы, в том числе, схема баз данных и, в-третьих, разрабатывается система процедур, реализующих все транзакции системы.

Структурную модель системы в технологии ГП можно отнести к классической архитектуре «трехуровневый клиент-сервер». Однако иногда приходится вводить дополнительные уровни программных модулей, так что реально можно говорить о многоуровневой архитектуре клиент-сервер. Клиентские рабочие места работают в режиме удаленного доступа и общаются с серверами по определенному протоколу сетевых каналов. Серверные компоненты взаимодействуют с СУБД и организуют очередность выполнения транзакций. Это важно для многопользовательских систем реального времени.

Рассмотрим основные модельные сущности ГП на примере САПР «ПЕЛЕНА-М», которая была разработана в ВЦ РАН по заказу ОКБ Сухого.

2. Система автоматизированного весового проектирования «ПЕЛЕНА-М»

САПР «ПЕЛЕНА-М» [2] предназначена для расчета массово-инерционных характеристик самолета с переменной массой, т.е. самолета с различными вариантами снаряжения и полезной нагрузки и переменным запасом топлива. У этой системы есть своя история. Впервые подобная программа была разработана и эксплуатировалась еще на ЭВМ БЭСМ-6. Тогда эта программа была названа ПЕЛЕНА. Она долгое время была основным инструментом для расчета центровочных зависимостей в ОКБ Сухого. Разумеется, программа ПЕЛЕНА за свою более чем тридцатилетнюю историю много раз модифицировалась, перерабатывалась для разных ЭВМ и ОС. Некоторое время назад у заказчика возникла идея кардинально изменить эту программу и постараться использовать современные информационные технологии и новые возможности пользовательского интерфейса. При этом было пожелание сохранить, по мере возможностей, математические модели, положенные в основу программы ПЕЛЕНА. В ВЦ РАН с помощью инструментального комплекса ГП была разработана практически новая система, которую, отдавая дань первому ее прототипу, назвали ПЕЛЕНА-М, Эта система сейчас входит в состав интегрированной автоматизированной системы весового проектирования, разрабатываемой в ВЦ РАН. Ниже на примере системы ПЕЛЕНА-М будут продемонстрированы ключевые сущности описания этого проекта средствами ГП.

project *pelenam /title="ПЛЕНА-М 2015" /version="03.09".*

package *privpln //Описание совокупности прав и привилегий пользователей системы.*

package *pln //В этом пакете дано описание основных типов и струткур данных проекта.*

genbd *gpln // Логическая схема БД. В ГП реализована встроенная СУБД сетевого типа.*

package *eval // Описание процедур, связанных с расчетом МИХ самолета.*

crpgate *pelenacrp //Декларация сервера безопасности.*

port *pelenaprt // Спецификации порта связи клиентского модуля с прикладным сервером*

server *pelenasrv // Описание процедур выполнения запросов, специфицированных в pelenaprt.*

window *w_plane // В этом и последующих пакетах типа* ***window*** *и* ***wintable*** *дано описание оконного интерфейса системы.*

dialog *d_plane // В этом и последующих пакетах типа* ***dialog*** *даны описания ползовательских диалогов для ввода информации.*

application *pelenausr // Описание клиентского модуля расчета МИХ ЛА с переменной массой.*

application *catalogusr // Описание модуля ведения в БД каталога элементов нагрузки.*

sysadm *pelenacrp //Декларация модуля администратора безопасности.*

На Рис. 1 показана логическая схема системы ПЕЛЕНА-М.

Заметим, что организация доступа к системе – система безопасности (pelenacrp) – формируется полностью автоматически, поскольку ее структуры практически никак не зависят от прикладной задачи. Единственная естественная связь осуществляется через систему привилегий (***package*** *privpln*), которая определяет набор прав доступа к информации для разных групп пользователей. Перечень привилегий задается при описании проекта, а предоставление их конкретным пользователям осуществляет администратор системы безопасности с помощью модуля **sysadm_pelenacrp**, который также создается автоматически. Вся информация о пользователях системы хранится в **genbd pelenacrp,** программное обеспечение для работы с которой также генерируется средствами ГП. В другой базе данных **genb gpln** хранится вся содержательная прикладная информация системы.

Вообще говоря, любой проект ИВС начинается с описания предметной области, т.е. с описания структуры данных, логической схемы хранения информации, отношений между основными объектами предметной области. В проекте ПЕЛЕНА-М описание предметной области в основном сконцентрировано в пакетах pln.pkg и gpln.gbd. Некоторые структуры данных описаны в других местах, но системообразующая информация о предметной области дана в пакете БД.

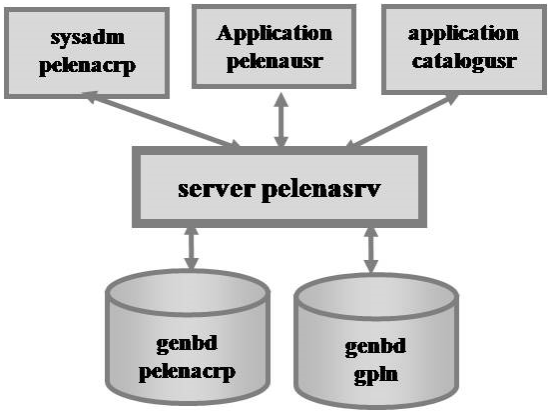

Рис. 1. Структура системы ПЕЛЕНА-М

Сетевая БД состоит из записей (кортежей) и отношений между ними.

record *obj: pln.t_obj; // кортеж параметров снаряженного и нагруженного самолета.*

record *plane: pln.t_plane; // кортеж параметров пустого самолета.*

record *crew: pln.t_crew; // кортеж параметров варианта снаряжения самолета.*

record *fuel: pln.t_fuel; // кортеж параметорв и характеристик текущего запаса топлива.*

record *equip: pln.t_equip; // кортеж параметров варианта загрузки самолета.*

record *unit: pln.t_unit; // кортеж параметров элементов полезной нагрузки.*

Формат описания связей между записями БД, т.н. наборов, принятый в сетевых БД, имеет следующий вид:

set list_obj **member** obj**;** // список вариантов изделий, хранящихся в БД,

set list_unit **member** unit**;** // список элементов полезной нагрузки,

set obj_plan **owner** obj **member** plane**;** // различные модификации пустого изделия,

set obj_crew **owner** obj **member** crew**;** // варианты снаряжения для заданного самолета,

set obj_fuel **owner** obj **member** fuel**;** // варианты программ выработки топлива в полете,

set obj_equip **owner** obj **member** equip**;** // варианты загрузки самолета,

set equip_unit **owner** equip **member** unit**;** // список элементов нагрузки выбранного варианта.

Здесь приведены лишь основные элементы модели предметной области. В проекте ПЕЛЕНА-М общее число объектов этой модели намного больше, но большинство из них связано с деталями описания и многими нюансами решаемых задач. Главное, что нужно подчеркнуть, что для программной реализации всех манипуляций и расчетов даже с таким небольшим набором данных для создания новых экземпляров описанных объектов, для их размещения в БД, для последующего поиска, модификации и проведения необходимых расчетов потребуется написать большой объем программного кода. Технология ГП позволяет существенно сократить трудоемкость ручного написания программ. Приведенные выше фрагменты формального описания проекта и другие элементы описания, о которых будет сказано ниже, обладают достаточной полнотой и точностью, чтобы можно было, абстрагируясь от содержательных задач, основываясь только на формальных моделях, автоматически конструировать другие модели и алгоритмы для построения необходимого программного кода.

Одна из задач, которую решает ГП, это организация запросов от пользователей системы к серверу и к базе данных. В технологии ГП запрос (ключевое понятие **request)** занимает одно из центральных мест. Все запросы в ГП должны быть специфицированы с указанием структур передаваемых и получаемых данных. Технологическим протоколом обмена в ГП является протокол TCP/IP. В описании проекта описываются именованные порты, через которые осуществляются взаимодействие клиентских приложений с серверными компонентами. Запросы, приписанные к одному порту, специфицируются в соответсвующем пакете типа **port**. Формат спецификаций запросов имеет следующий вид:

request <имя запроса> **[input (**<входные параметры>**)] [: output(**<выходные параметры>**)];**

Формируются запросы в клиентских приложениях. Описание клиентских модулей дается в пакетах типа **application**. В системе ПЕЛЕНА-М два прикладных клиентских модуля и один модуль системного администратора. Модуль системного администратора формируется автоматически, и в проекте он лишь декларируется. Прикладные же модули описываются разработчиками системы. Ниже приведена структура описания одного из декларированных прикладных модулей:

application pelenausr: "РАСЧЁТНЫЙ МОДУЛЬ ПЕЛЕНА-2015" /icon16=pelenausr

global ():(pln.t_bool global_cent) // список глобальных переменных, доступных в описании

client pelenaclt **port** pelenaprt //ID клиента в сети и указание рабочего сетевого порта

<Раздел описания пользовательского оконного интерфейса приложения>

<Раздел описания локальных процедур>

<Раздел описания пользовательских команд данного приложения >

<Описание сценариев работы в приложении: меню, горячие клавиши, кнопки,… >

Количество окон в клиентском приложении не ограничено.

На Рис. 2. показан пример реализации окон в проекте ПЕЛЕНА-М.

Стрелками на рисунке показаны возможные переходы между окнами. Переход и передача параметров между ними определяется специальными процедурами, описанными в разделе команд. Структура окон и их параметры задаются в пакете *.app клиентского приложения, а формы представления в них содержательной информации и допустимые манипуляции с данными окон описываются в отдельных пакетах типа **window** и **wintable**.

Запросы клиентского модуля формируются, как операторы в теле команды.

command *<имя команды>* ***layout*** *<имя окна>:<"название команды">*

*{… <имя порта>**.**<имя запроса >**(**<сетевые параметры клиента>**,**<параметры запроса>**); …** }*

Имя и типы фактических параметров запроса должны однозначно соответствовать спецификациям, приведенным в пакете соответствующего порта.

Исполнение запроса осуществляется сервером, которому он адресован. В процессе исполнения сервер обращается к своим БД. «Свои БД» указываются в пакете описания соответсвующего сервера. Например, в проекте ПЕЛЕНА-М у сервера pelenasrv.srv своими БД являются сетевая БД *gpln* и сетевая БД *pele-*

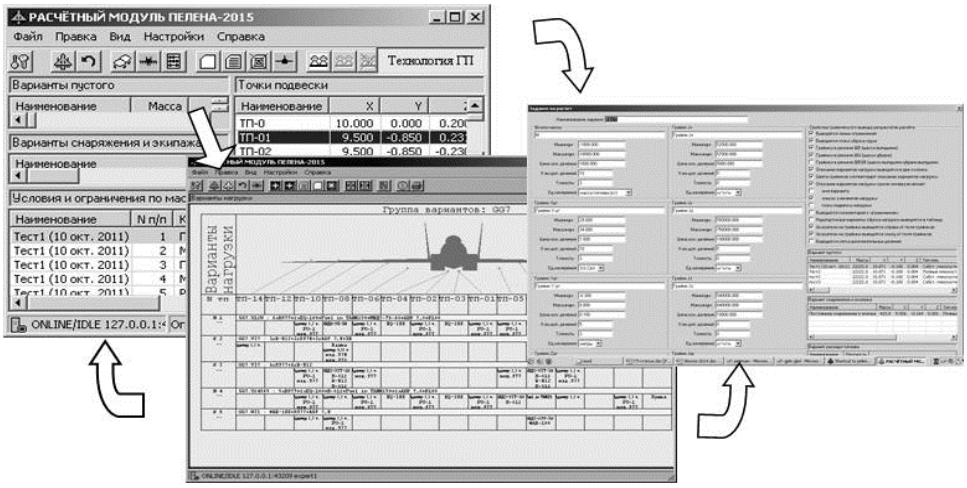

Рис. 2. Оконный пользовательский инерфейс в системе ПЕЛЕНА-М

nacrp: Ниже приведена структура описания этого сервера:

server *pelenasrv:"СЕРВЕР ПЕЛЕНА" /icon16=pelenasrv*

genbd *gpln:gpln // ссылка на прикладную сетевую БД проекта.*

crpgate *pelenacrp <параметры соединения сервера с БД системы безопасности >*

genbd *pelenacrp // ссылка на сетевую БД системы безопасности (БД СБ)*

<Раздел описания локальных процедур сервера>

<Раздел описания процедур выполнения клиентских запросов>

Описание процедур выполнения запросов имеет следующий вид:

request *<имя порта>.<имя запроса>* ***[input (****<вх. параметры>****)]:[output(****<вых. параметры>****)]***

{ *<тело процедуры выполнения запроса>* ***}***

Краткое описание сложного проекта не может дать полного представления о разработанной системе. На Рис. 3 показана схема обработки исходного описания проекта и автоматической генерации программного кода системы.

Об оценке эффективности применения ГП можно судить по соотношению объема исходного описания к объему сгенерированного С-кода системы. Общий объем исходного описания проекта ПЕЛЕНА-М составляет 0.7 МБ (64 файла). Полный комплект сгенерированных С-текстов проекта ПЕЛЕНА-М составил 7.3 МБ (657 файлов). Таким образом, применение технологии ГП позволило более чем в десять раз сократить трудозатраты при разработке этого проекта. Конечно, всякая количественная оценка эффективности весьма условна, но качественное преимущество разработанной технологии неоспоримо и состоит в том, что, во-первых, разработчик системы занимается в основном содержательными задачами и не отвлекается на вопросы системной организации программ, которые представляют собой сложный клубок компьютерных техник. Во-вторых, ГП как инструментальный комплекс берет на себя задачу валидации реализованных программ, которая обеспечивается строгим контролем соответствия типов данных формальных и фактических параметров процедур, контроля полноты и непротиворечивости конструкций, недопустимости избыточности программного кода. В-третьих, применение ГП на этапе эксплуатации разработанных программ позволяет существенно сократить затраты на их сопровождение.

Рис. 3. Последовательность автоматической генерации программного кода проекта

3. Автоматизированная система бюджетного управления корпорацией

Рассмотрим еще одну ИВС, реализованную с помощью Генератора проектов. Эта система относиться к так называемым АСУ, автоматизированным системам управления. В данном случае речь идет о бюджетном управлении корпорацией. Бюджетное управление - это технология управления, которая позволяет руководству ставить перед своими структурными подразделениями и дочерними предприятиями конкретные производственные задачи посредством бюджетных ограничений и осуществлять постоянный контроль над выполнением этих ограничений. Автоматизированная система бюджетного управления (АСБУ [3]) предназначена для информационной и аналитической поддержки решения основных задач бюджетного управления. Такими задачами являются: планирование бюджетных ограничений и основных финансово–экономических показателей для дочерних предприятий, сбор информации о фактически реализованных значениях финансово–экономических показателей за отчетные периоды, анализ выполнения бюджетов и планов, составление сводных и аналитических отчетов, оценка эффективности вложения средств.

Структурно-параметрическая информационная модель предметной области АСБУ в данном проекте, в отличие от системы ПЕЛЕНА-М, реализована в рамках реляционной БД. Основными объектами этой модели являются:

- каталог финансово-экономических показателей корпорации,
- формы плановых, отчетных и сводных показателей предприятий корпорации,
- реестр значений плановых и отчетных показателей предприятий корпорации.

Каталог финансово-экономических показателей кроме позиций, связанных с планом бухгалтерских счетов, содержит множество показателей по всей номенклатуре выпускаемой продукции как в рублевых, так и в натуральных единицах, а также множество показателей эффективности экономической деятельности. Важной особенностью этого каталога является обязательная кодификация показателей. Каталог является открытой структурой, которая может дополняться и модифицироваться в процессе эксплуатации АСБУ. В составе каталога можно вводить различные агрегированные показатели, причем алгоритмы агрегирования могут вводить пользователи системы по своему усмотрению. Благодаря гибкости и широким возможностям ставить различные задачи, АСБУ стала исследовательским инструментом работников подразделений, занимающихся контролем, анализом и планированием в корпорации. Одной из важных особенностью АСБУ стало то, что появилась возможность оперативной связи с дочерними предприятиями. На каждом из них был установлен свой экземпляр АСБУ, и была обеспечена электронная связь между дочерними предприятиями и центральным офисом корпорации. Это позволило им общаться в одних и тех же терминах, поскольку на предприятиях корпорации действовал тот же каталог показателей, что и в плановом управлении корпорации. Естественно, на дочерних предприятиях могли учитывать свою специфику и свою номенклатуру продукции.

Обмен информацией между предприятиями корпорации выполнялся в виде согласованных документов, формы которых настраивались и утверждались на уровне руководства корпорации. Перечень и структура форм представления данных также открыты для дополнений и модификации. В АСБУ были введены несколько форм представления информации – плановые формы, отчетные, сводные, аналитические за разные отчетные периоды. Экономические показатели в формах маркировались специальными признаками – плановые показатели, фактические данные, предварительные результаты, краткосрочные прогнозы и др. Совокупность действующих форм является не набором разрозненных документов, а представляет собой единое поле, отражающее финансово-экономическую деятельность корпорации. АСБУ осуществляет контроль за полнотой и непротиворечивостью информации во всей совокупности согласованных документов. На Рис. 4 показана структура АСБУ и ее применения на предприятиях корпорации.

На всех предприятиях корпорации было установлено нескольких экземпляров одной и

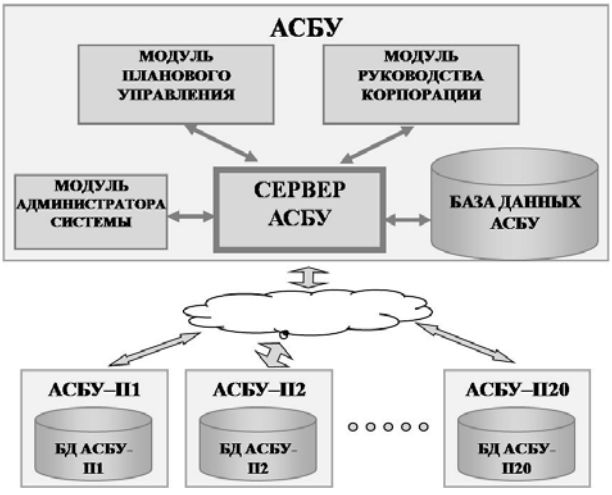

Рис. 4. Система бюджетного управления корпорации

той же системы, но по-разному конфигурированной. Фактически в корпорации и в ее дочерних предприятиях работало более двух десятков экземпляров АСБУ, каждая со своим сервером и со своей базой данных. Связь между ними осуществлялась путем обмена документами в режиме off-line. О возможности по-разному конфигурировать АСБУ заявлено в заголовке проекта:

project *budget /title="Автоматизированная система бюджетного управления (АСБУ)"*

// для центра конфигурационная переменная bz=true, для дочерних предприятий bz=false.

Такая схема «клонирования» системы АСБУ оказалась весьма практичной и удобной. Нами рассматривалась и другая конструкция проекта с удаленными рабочими местами на дочерних предприятиях, но поскольку обмен между «центром» и «периферией» был организован на уровне документов, то взаимодействие в режиме on-line было излишне. При такой автономизированной схеме, тем не менее, вся информация о деятельности дочерних предприятий поступала в центр, анализировалась и сохранялась в центральной базе данных. Центральное хранилище информации о деятельности корпорации за десять лет функционирования накопило богатейший материал для проведения различных аналитических исследований. В АСБУ были заложены механизмы для регрессионного анализа тенденций развития корпорации, для составления прогнозных и аналитических форм. Наличие структурированного и кодифицированного каталога упорядочивает всю информацию в базе данных и в архивах, позволяет быстро находить данные по всем направлениям экономической деятельности корпорации. Насколько мы понимаем, именно такой подход к планированию и управлению сейчас называют цифровой экономикой.

Заключение

Генератор проектов, представленный в настоящей работе, является развивающейся системой [4]. Разумеется, технологическая основа ГП – проектный подход и генерация программного кода – остаются незыблемыми, тем более, что они оправдали себя при разработке реальных ИВС. Сейчас идет активная работа над новейшей версией ГП. В связи с прошедшей реорганизацией, в ВЦ РАН эта тема закрыта. В этом заключении мы дадим перечень тех основных проектов, которые были реализованы в разное время с помощью ГП.

Система весовых расчётов и весового контроля пустого ЛА – АСВР [5]. (1980-2015 гг. ОКБ Сухого). Эта система предназначена для построения весовой модели конструкции летательных аппаратов в целях проведения весовых расчетов и весового контроля. Расчеты производятся как по древовидной структуре конструктивного членения объекта, так и с учетом принадлежности отдельных деталей различным подсистемам. Система используется на всех этапах жизненного цикла изделия: на различных стадиях проектирования, на стадии опытного и серийного производства, на стадии эксплуатации. За более чем тридцатилетнюю историю использования система несколько раз обновлялась и до сих пор сопровождается авторами разработки.

Банковская система ГАМБИТ [6]. (1995-2000 гг. Сбербанк РФ). Система комплексной автоматизации многофилиального универсального коммерческого банка, была внедрена и в течение нескольких лет эксплуатировалась в Башкирском банке Сбербанка России. Система автоматизировала деятельность Главной бухгалтерии банка, ОПЕРУ, Территориального расчетного центра и отделений банка. Впервые в отечественной практике была построена централизованная система для крупного банка республиканского масштаба с развитой филиаль-

ной сетью. ГАМБИТ существенно отличалась от других автоматизированных систем банковской автоматизации обработкой всех банковских операций в режиме on-line, а также постоянной поддержкой бухгалтерского баланса банка в актуальном состоянии.

Система приёма банковских платежей MassPay [7]. (2005-2015 гг. Сбербанк РФ). Система предназначена для приема широкого спектра платежей населения (коммунальные, налоговые, оплата товаров и услуг и т.п.) на банковских устройствах самообслуживания - банкоматах и инфокиосках с помощью банковских карт и наличных. Центральной компонентой Системы является фронтальная подсистема, непосредственно управляющая диалогом пользователя с устройством самообслуживания. Фронтальная подсистема взаимодействует в режиме on-line с произвольными внешними биллинговыми центрами.

Банковская биллинговая система «Енисей» [8]. (2005-2015 гг. Сбербанк РФ). Биллинговая система предназначена для эксплуатации в банках в целях построения автоматизированных комплексов электронного банковского самообслуживания. В основе концепции БС «Енисей» лежит понятие профиля клиента, в котором сосредоточены описания ряда операций, которые клиент может выполнять с помощью Системы. Спектр этих операций довольно широк и определяется банком, эксплуатирующим систему. Это платежи за коммунальные и телекоммуникационные услуги, платежи по ссудам физических лиц, банковские переводы, налоговые платежи, оплата железнодорожных, авиационных и театральных билетов, штрафы и госпошлины и т.д. Доступ клиентов к их профилям и, соответственно, выполнение операций осуществляется по различным каналам: через сеть устройств банковского самообслуживания, через Интернет; с помощью GSM-телефонов. БС «Енисей» может взаимодействовать с внешними автоматизированными системами (платежи в пользу поставщиков услуг) и с подсистемами АБС (банковские операции) в режимах off-line и on-line.

Система моделирования нейронных структур МОЗЖЕЧКА [9]. (2006-2015 гг. грант РФФИ). Инициативная работа, которая направлена на моделирование нейронных структур мозга и изучение возможности использования построенных моделей для решения прикладных задач управления. Был разработан научно-исследовательский программный стенд, с помощью которого генерировалась модель нейронной структуры мозжечка, включённая во внешний контур управления. На этом стенде изучалось поведение таких систем.

Автоматизированная система весового проектирования самолетов - АСВП [10]. (2016-2018 гг.) Данная система разрабатывалась как объединение нескольких ранее разработанных систем весового проектирования – системы весовых расчетов и весового контроля пустого самолета, снаряженного и загруженного самолета, программы тарировки топливных баков, системы анализа центровки самолета в процессе выработки топлива. Система разрабатывается как интерактивная многопользовательская информационная система клиент-серверной архитектуры с централизованной базой данных. Информационным ядром и основой АСВП является единая структурно-параметрическая весовая модель самолета. АСВП разрабатывается с помощью инструментального комплекса «Генератор проектов».

Литература

1. Л.Л. Вышинский, И.Л. Гринев, Ю.А. Флеров, А.Н. Широков, Н.И. Широков. Генератор проектов – инструментальный комплекс для разработки «клиент - серверных» систем // Информационные технологии и вычислительные системы. 2003, № 1-2. С. 6-25
2. Л.Л. Вышинский, Н.И. Широков. Система автоматизации расчетов массово-инерционных характеристик ЛА с переменной массой. // Сборник статей «Развитие и применение инструментального комплекса ГЕНЕРАТОР ПРОЕКТОВ». Отв. ред. Ю.А. Флеров. М.: ВЦ РАН, 2014. С. 20-31.
3. Л.Л. Вышинский. Реализация автоматизированной системы бюджетного управления средствами Генератора проектов. // Сборник статей «Автоматизация проектирования инженерных и финансовых информационных систем средствами ГЕНЕРАТОРА ПРОЕКТОВ». Отв. ред. Ю.А. Флеров. М.: ВЦ РАН, 2010. С. 68-79.
4. А.А. Логинов, Н.И. Широков. Новые возможности Генератора проектов. // Сборник статей «Развитие и применение инструментального комплекса ГЕНЕРАТОР ПРОЕКТОВ». Отв. ред. Ю.А. Флеров. М.: ВЦ РАН, 2014. С. 6-19.
5. Н.И. Широков. Автоматизированная система весовых расчетов в САПР ЛА. // Сборник статей «Автоматиза-

ция проектирования инженерных и финансовых информационных систем средствами ГЕНЕРАТОРА ПРОЕКТОВ». Отв. ред. Ю.А. Флеров. М.: ВЦ РАН, 2010. С. 55-67.
6. Л.Л. Вышинский, И.Л. Гринев, В.П. Катунин, И.В. Лабутин, Ю.А. Флеров, Н.И. Широков. Банковские информационные технологии. Часть I. Часть I I. М.: Вычислительный центр РАН, 1999. 263 с.
7. И.Л. Гринев, А.А. Логинов, Н.И. Широков, А.Н. Широков. Система банковского самообслуживания. // Сборник статей «Автоматизация проектирования финансовых информационных систем». Отв. ред. Ю.А. Флеров, Л.Л. Вышинский. М.: ВЦ РАН, 2004. С. 87-94.
8. А.Н. Широков, А.А. Логинов, И.Л. Гринёв. Организация функционирования биллинговой системы в автоматизированной банковской среде. // Сборник статей «Автоматизация проектирования инженерных и финансовых информационных систем средствами ГЕНЕРАТОРА ПРОЕКТОВ». Отв. ред. Ю.А. Флеров. М.: ВЦ РАН, 2010. С.45-54.
9. Л.Л. Вышинский. Программный комплекс для изучения нейрокомпьютерной модели мозжечка. // Сборник статей «Развитие и применение инструментального комплекса ГЕНЕРАТОР ПРОЕКТОВ». Отв. ред. Ю.А. Флеров. М.: ВЦ РАН, 2014. С.60-68.
10. Л.Л. Вышинский, Ю.А. Флеров, Н.И. Широков. Автоматизированная система весового проектирования самолетов. // Информатика и её применения. 2018. Т. 12. Вып. 1. С. 18-30.

Флеров Юрий Арсениевич. Вычислительный центр им. А.А. Дородницына Федерального государственного учреждения "Федеральный исследовательский центр "Информатика и управление" Российской академии наук" г. Москва, Россия. Зам. директора, чл.-корр. РАН, доктор физико-математических наук, профессор. Количество печатных работ 176 (в т.ч. 8 монографий). Область научных интересов: математическое моделирование, информатика, автоматизация проектирования. E-mail: fler@ccas.ru

Вышинский Леонид Леонидович. Вычислительный центр им. А.А. Дородницына Федерального государственного учреждения "Федеральный исследовательский центр "Информатика и управление" Российской академии наук" г. Москва, Россия. И.о. ведущего научного сотрудника, кандидат физико-математических наук. Количество печатных работ: 56. Область научных интересов: информатика, автоматизация проектирования. E-mail: wyshinsky@mail.ru

Computer-aided design of applied information computing systems

Yu.A. Flerov, L.L. Vyshinsky

Institution of Russian Academy of Sciences Dorodnicyn Computing Centre of RAS, Moscow, Russia

In the article the questions of automation of designing and development of complex applied information-computing systems are considered. The technology of designing applied software systems is briefly described, based on the project approach and the tool complex Project Generator. On the example of concrete application systems, the main provisions of this approach are described. In the Computing Center of the Russian Academy of Sciences, a number of large information systems were developed in different years with the help of the Project Generator. The article lists some of them, their characteristics are given and the efficiency of using the Project Generator technology is shown.

Keywords: math modeling, design automation, project generator, code generation, client-server architecture.

DOI 10.14357/20718632180303

References

1. L.L. Vyshinskiy, I.L. Grinev, Yu.A. Flerov, A.N. Shirokov, N.I. Shirokov. 2003. Generator proektov – instrumental'nyy kompleks dlya razrabotki «klient - servernykh» sistem. [Information technology and computer systems] 1-2: 6-25.
2. L.L. Vyshinskiy, N.I. Shirokov. 2014. Sistema avtomatizatsii raschetov massovo-inertsionnykh kharakteristik LA s peremennoy massoy. Sbornik statey «Razvitie i primenenie instrumental'nogo kompleksa GENERATOR PROEKTOV». Otv. red. Yu.A. Flerov [Computing Center of the RAS]: 20-31.
3. L.L. Vyshinskiy. 2010. Realizacija avtomatizirovanoy sistemi bjudgetnogo upravlenija sredstvami Generatora proektov. Sbornik statey «Avtomatizacija proektirovanija ingenernyh i finansovyh informacionnyh system sredstvami GENERATORA PROEKTOV». Otv. red. Yu.A. Flerov [Computing Center of the RAS]: 68-79.
4. A.A. Loginov, N.I. Shirokov. 2014. Novye vozmognosti Generatora proectov. Sbornik statey «Razvitie i primenenie instrumental'nogo kompleksa GENERATOR PROEKTOV». Otv. red. Yu.A. Flerov [Computing Center of the RAS]: 6-19.
5. N.I. Shirokov. 2010. Avtomatizirovanaja sistema wesovych raschetov v SAPR LA. Sbornik statey «Avtomatizacija proektirovanija ingenernyh i finansovyh informacionnyh system sredstvami GENERATORA PROEKTOV». Otv. red. Yu.A. Flerov [Computing Center of the RAS]: 55-67.

6. L.L. Vyshinskiy, I.L. Grinev, W.P. Katunin, I.V. Labutin, Yu.A. Flerov, N.I. Shirokov. 1999. Bankovsie informacionnye technologii. Ch. I, Ch. II. [Banking information technologies. P. I, P. II]. Moscow: CCAS. 263 p.
7. I.L. Grinev, A.A. Loginov, N.I. Shirokov, A.N. Shirokov. 2004. Sistema bankowskogo samoobslugivanija. Sbornik statey «Avtomatizacija proektirovanija finansovyh informacionnyh system. Otv. red. Yu.A. Flerov, L.L. Vyshinskiy/ [Computing Center of the RAS]: 87-94.
8. A.N. Shirokov, A.A. Loginov, I.L. Grinev. 2010. Organizacija funkcionirovanija billingovoj sistemy v avtomatizirovanoj bankovskoj srede. Sbornik statey «Avtomatizacija proektirovanija ingenernyh i finansovyh informacionnyh system sredstvami GENERATORA PROEKTOV». Otv. red. Yu.A. Flerov [Computing Center of the RAS]: 45-54.
9. L.L. Vyshinskiy. 2014/ Programnyj kompleks dlja izuchenija nejrokompjuternoj modeli mozgechka. Sbornik statey «Razvitie i primenenie instrumental'nogo kompleksa GENERATOR PROEKTOV». Otv. red. Yu.A. Flerov [Computing Center of the RAS]: 60-68.
10. L.L. Vyshinskiy, Yu.A. Flerov, N.I. Shirokov. 2018. [Computer-aided design of applied information computing systems] Inform. Appl. 1:18-30.

Flerov Yuri Arsenievich Corresponding Member of RAS, Doctor of Science in physics and mathematics, professor, Deputy Director, Institution of Russian Academy of Sciences Dorodnicyn Computing Centre of RAS

Vyshinsky Leonid Leonidovich Candidate of Sciences (PhD) in physics and mathematics, Leading Researcher/, Institution of Russian Academy of Sciences Dorodnicyn Computing Centre of RAS.