

УДК 681.3.08

Динамическая отработка программного обеспечения бортовых цифровых вычислительных машин систем управления объектов ракетно-космической техники

Я. Е. Айзенберг, А. В. Бек, Ю. М. Златкин,
В. П. Каменев, Б. М. Конорев, В. Т. Щербаченко

АО ХАРТРОН, Харків

Надійшла до редакції 26.11.96

На якісному рівні визначені мета та значимість динамічного відпрацювання і атестації програмного забезпечення (ПЗ) БЦОМ у формі «електронних пусків» в контексті загальної технології створення систем керування (СК). Неформально викладена методологія динамічного відпрацювання ПЗ БЦОМ, що розглядається як дуже важлива складова прикладної теорії програмування БЦОМ СК ракетно-космічної тематики, що відноситься до класу систем жорсткого реального часу. В термінах потокової моделі розглянуті базові схеми динамічного відпрацювання ПЗ БЦОМ і сформульовані основні концепції їх інструментальної підтримки в CASE-середовищах (Computer Aided Software Engineering): концепції зв'язаних цільових контурів, синхронізації даних і адаптивного керування технологічними процесами. Представлено узагальнення досвіду робіт АО ХАРТРОН, починаючи з 1960-х рр., в області створення ПЗ вбудованих БЦОМ СК за профілюючими напрямками ракетно-космічної тематики: важка ракета-носій «Енергія», сімейство функціональних модулів «Квант», «Кристал», «Спектр», «Природа» орбітальних пілотованих станцій, сімейство міжконтинентальних балістичних ракет з розділовими головними частинами, включаючи ракети типів SS18 і SS19, транспортний корабель постачання і енергетичний функціонально-вантажний модуль американо-російської програми міжнародної космічної станції «Альфа» та ін. Коректність і практична ефективність концепцій підтверджена позитивними результатами динамічного відпрацювання ПЗ БЦОМ при забезпеченні понад 300 пусків (циклів натурних випробувань) ракет і космічних апаратів різних типів.

ВВЕДЕНИЕ

Для широкого спектра типов ракет и космических аппаратов программное обеспечение (ПО) бортовых цифровых вычислительных машин (БЦВМ) как элемент системы управления (СУ), являясь эквивалентной формой представления (реализацией) алгоритмов управления, в значительной мере определяет логику, динамику и точность функционирования объекта управления. Взаимодействие ПО БЦВМ с физическим оборудованием в реаль-

ном времени непосредственно в контурах управления сложнопротекающими необратимыми процессами в основных режимах работы объекта предопределяет предельно жесткие требования к качеству его отработки и высокую трудоемкость. Результатом этого является существование критической зависимости качества, сроков и стоимости СУ в целом от ПО БЦВМ. Основным направлением решения этой проблемы является использование адекватной технологии программирования БЦВМ.

Статья посвящена неформальному изложению

методических основ технологии и концепций построения инструментальных сред отработки управляющих программ СУ объектов ракетно-космической техники. Обсуждаемые проблемы входят составной частью в прикладную теорию программирования БЦВМ (встроенных ЭВМ). Представленные результаты являются обобщением опыта АО ХАРТРОН, начиная с 1960-х годов по настоящее время (Айзенберг и др., 1964–1996), в создании ПО БЦВМ для широкого спектра типов объектов управления ракетно-космической тематики, включающего:

- тяжелую ракету-носитель «Энергия»;
- семейство функциональных модулей орбитальной станции «Мир» («Квант», «Кристалл», «Спектр», «Природа»);
- семейство межконтинентальных баллистических ракет с разделяющимися головными частями, в том числе ракеты типов SS18, SS19 и др.

Отличительной чертой разработанной технологии динамической отработки ПО БЦВМ, наряду с высокой эффективностью в обеспечении качества, является ее высокая экономичность по отношению к вычислительным ресурсам инструментальных ЭВМ. Полномасштабное использование технологии возможно при ресурсовооруженности (по суммарной производительности и объемам ЗУ всех типов) инструментальных ЭВМ, в 10–100 раз более низкой, чем аналогичные работы в США.

Приведена типовая для рассматриваемой прикладной области архитектура и спецификация функциональных возможностей инструментальной среды отработки ПО, относящейся к классу CASE-сред, которая может рассматриваться как рекомендация для практической реализации.

Изложенные концепции в настоящее время используются ХАРТРОНОМ — единственной организацией в Украине, имеющей практический опыт программирования БЦВМ объектов ракетно-космической техники, — при разработке ПО БЦВМ энергетического функционально-грузового блока и транспортного корабля снабжения американо-российской программы создания международной космической станции «Альфа».

РОЛЬ И ЗАДАЧИ ДИНАМИЧЕСКОЙ ОТРАБОТКИ ПО БЦВМ В ОБЩЕЙ ТЕХНОЛОГИИ СОЗДАНИЯ СУ

Технология программирования БЦВМ, охватывающая область разработки управляющих алгоритмов и их программной реализации на БЦВМ, критическим образом определяет возможности реализации

требуемых характеристик, качество, сроки и стоимость создания СУ и объекта управления в целом. Уровень используемой технологии является решающим фактором достижения конкурентоспособности организаций, разрабатывающих СУ.

Центральное место в технологии программирования БЦВМ занимают проблемы наземной отработки. Наиболее трудоемкая среди них — динамическая отработка. В понятие динамической отработки входит проверка работоспособности ПО при имитации внешних условий, максимально приближенных к реальным, и устранение замеченных ошибок. Основным результатом и завершающим актом динамической отработки является аттестация качества ПО.

По существу объектом динамической отработки являются как управляющие алгоритмы (например, в форме общематематических нотаций, представляющих спецификации для программирования), так и их программная реализация на БЦВМ. Поэтому в общем случае при динамической отработке используются два критерия: корректность управляющих алгоритмов и взаимно однозначное соответствие управляющих алгоритмов и ПО БЦВМ. Эти критерии могут использоваться в различных сочетаниях в технологических процессах динамической отработки.

Динамическая отработка может проводиться как автономно для подсистем (например стабилизации), так и для режимов работы СУ в целом (например выведение,стыковка на орбите и т. п.). Последний случай представляет комплексную динамическую отработку. Целесообразное сочетание объемов и видов динамической отработки определяется характеристиками конкретного объекта управления.

Динамическая отработка ПО БЦВМ неразрывно связана с технологией разработки СУ в целом, в процессе которой формируются требования к ПО как элементу СУ, этапам его разработки, верификации и сопровождения.

Особая роль технологии и инструментальных средств динамической отработки ПО БЦВМ в общей технологии создания СУ определяется тем, что основные характеристики СУ и объекта управления в различных режимах работы, включая полетные, могут быть опосредованы через значения переменных управляющей программы, реализуемой БЦВМ. При натурных испытаниях ПО БЦВМ является одним из основных источников телеметрической информации о характеристиках и поведении объекта управления.

Гибкость и адаптивность, присущие программам ЭВМ вообще, при разработке СУ объектов ракетно-

космической тематики рождают многочисленные и вполне обоснованные «состязания» решать возникающие проблемы, связанные с неизбежными на этапах отработки ошибками в ИД, уточнениями характеристик, доработками аппаратуры СУ и объекта управления, путем соответствующих коррекций ПО. Реально такие проблемы существуют вплоть до прекращения эксплуатации объекта управления и обязательным условием их решения является проведение в необходимом объеме динамической отработки для верификации ПО БЦВМ после внесения в него изменений.

Еще одна группа задач разработки и сопровождения СУ — выбор и верификация полетных заданий — основана на использовании технологии динамической отработки ПО. Основным способом решения этих задач является проведение циклов моделирования с реальным полетным заданием и имитацией условий, максимально приближенных к условиям конкретного запуска. Такие циклы моделирования иногда называются «электронными пусками».

В силу всего изложенного, технология и инструментальные средства отработки ПО БЦВМ должны быть спроектированы так, чтобы наряду с высокой эффективностью обеспечивалась перспектива их длительной эксплуатации и консервации в том числе и в расчете на возможную смену вычислительных платформ (типов инструментальных ЭВМ, операционных систем, СУБД и т. п.) в течение всего жизненного цикла объекта управления, продолжительность которого для некоторых типов объектов может составлять более 30 лет.

СПЕЦИФИЧЕСКИЕ ХАРАКТЕРИСТИКИ ПО БЦВМ КАК ОБЪЕКТА ТЕХНОЛОГИИ

СУ объектов ракетно-космической техники рассматриваемого спектра типов относятся к классу систем жесткого реального времени. ПО БЦВМ таких СУ является элементом, включенным непосредственно в контура управления сложнопротекающими необратимыми процессами. В силу этого характеристики ПО в значительной мере определяют логику, точность и динамику СУ в целом. А это предопределяет предельно высокие требования к его качеству, и прежде всего к надежности.

Для систем этого класса характерны циклическое решение задач управления с периодом квантования 10—100 мс и весьма высокие требования к скорости реакции на события, происходящие во внешней среде. Требуемое значение времен обработки прерываний и коммутации задач составляет единицы-

десятков мкс. Время используется как параметр вычислений в отдельных алгоритмах и как управляющий параметр при реализации полной совокупности асинхронных динамических процессов управления объектом, образованных большим числом независимых источников активности (датчиков). Общим требованием является недопустимость потерять заявок на решение задач и получение результатов решений в строго регламентированные отрезки реального времени, определяемые логикой и динамикой процессов управления объектом.

Ошибки учета реального времени при проектировании и разработке ПО БЦВМ относятся к наиболее трудно выявляемым и в значительной мере определяют общую высокую трудоемкость его создания.

В целом же ошибки в ПО БЦВМ связаны с потенциальной опасностью больших материальных потерь (аварии материальной части, задержки работ), а в ряде случаев непосредственно определяют безопасность работ с объектом управления. Положение усложняется недопустимостью отработки ПО при реальном использовании объекта. Это существенно повышает необходимость тщательного обоснования технологии и инструментальных средств отработки и аттестации ПО БЦВМ.

БАЗОВЫЕ СХЕМЫ ДИНАМИЧЕСКОЙ ОТРАБОТКИ

Схема на рис. 1 иллюстрирует укрупненно спецификацию инструментальных средств и потоки данных при использовании для отработки ПО двух базовых схем: разомкнутых и замкнутых (Айзенберг, Конорев, 1982).

Технологическая схема первого типа — разомкнутая — основана на независимой разработке по одним и тем же алгоритмам двух программ управления, реализуемых соответственно на встроенной (объектной) БЦВМ и инструментальной технологической ЭВМ. С точки зрения целей отработки первая реализация, ПО БЦВМ, интерпретируется как «фактическая», а реализация на инструментальной ЭВМ, т. е. модель управляющих алгоритмов, — как «эталонная» программа.

Фактическая и эталонная программы разрабатываются «в две руки» различными коллективами специалистов.

Эталонная реализация замыкается на модель объекта управления и исполняется для некоторого тестового набора исходных данных. При этом образуется два потока данных: от модели объекта к управляющей программе и от управляющей программы к модели объекта. Первый поток имеет

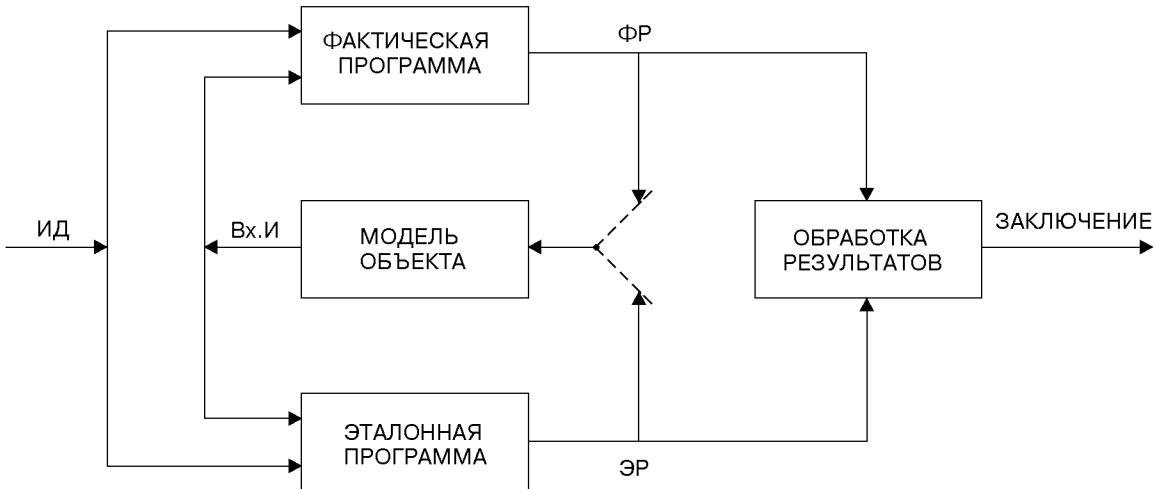


Рис. 1. Обобщенная схема отработки ПО

смысл входной информации (ВхИ) и исходных данных (ИД), а второй поток представляет эталонные результаты (ЭР), получаемые при реализации модели алгоритмов. Часть потока эталонных результатов, например результаты промежуточных вычислений, не используется моделью объекта и служит для обеспечения необходимого уровня диагностики при отладке ПО БЦВМ.

На следующем шаге фактическая управляющая программа — ПО БЦВМ исполняется от того же потока ВхИ при том же тестовом наборе ИД. Получаемый при этом поток результатов, включая промежуточные, интерпретируется как фактические результаты (ФР).

Метод отработки заключается в сравнении фактических и эталонных результатов. Совпадение их с точностью до некоторого допуска, определяемого различными погрешностями вычислений на инструментальной и объектной ЭВМ, является критерием соответствия обеих программ алгоритму. Это означает, что обработка результатов в технологической схеме I типа может производиться по формальным критериям, основанным на допусковом контроле фактических результатов отрабатываемой программы. Этalonные значения генерируются программой. Допуски на отличия эталонных и фактических результатов устанавливаются на основе анализа вычислительных погрешностей фактической и эталонной программ.

Главное достоинство этого метода заключается в высокой диагностической способности при отладке ПО. Недостатками его является высокая трудоемкость, связанная с необходимостью раздельной от-

ладки алгоритмов и ПО БЦВМ. Ошибки как в моделях алгоритмов, так и в ПО БЦВМ, приводят к несовпадению эталонных и фактических результатов и по определению должны быть устранены. Достоверность отработки определяется вероятностью совпадающих ошибок в эталонной и фактической программах.

Технологическая схема второго типа основана на оценке поведения модели объекта управления при взаимодействии ее с ПО БЦВМ (фактической управляющей программой) в замкнутой схеме с обратной связью. Достоинство этой технологической схемы заключается в уменьшении потребления ресурсов всех видов на одно испытание, так как эталонная программа не используется. Отработка алгоритмов и программ совмещена — модель объекта замкнута на фактическую (штатную) реализацию управляющих алгоритмов. Однако при этом локализация ошибок в ПО БЦВМ затруднена, так как отсутствуют программно генерируемые эталоны для фактических результатов.

Обработка фактических результатов в схеме этого типа производится на основе функционально ориентированных, «физических» критерии, всецело определяемых типом и характеристиками конкретного объекта управления. Такие критерии представляют конечные результаты работы СУ (например, точность выведения на орбиту, попадания в цель и т. п.), циклограммы функциональных и временных команд, логику функционирования и динамические погрешности на всех участках траектории и т. п.

Отработка в технологической схеме II типа при-

меняется после того, как доказано (с необходимой полнотой) соответствие управляющей программы алгоритмам управления с использованием схемы I типа.

С использованием описанных базовых схем могут быть реализованы следующие режимы отработки:

а) эталонный процесс замкнут, фактический отсутствует. Используется на этапах исследовательских задач и для отладки моделей объекта и алгоритмов;

б) фактический процесс разомкнут, эталонный замкнут. Типичная разомкнутая схема.

в) фактический процесс замкнут, эталонный отсутствует. Типичная замкнутая схема.

г) эталонный процесс разомкнут, фактический замкнут. Используется для специальных классов алгоритмов и исследовательских задач.

Для каждого из режимов возможны различные варианты организации во времени. Так, например, реализация моделей объекта, генерация эталонов, реализация ПО, обработка результатов могут производиться на инструментальной и объектной ЭВМ в общем случае в последовательно-параллельной схеме с различными вариантами совмещения процессов.

КОНЦЕПЦИИ ИНСТРУМЕНТАЛЬНОЙ ПОДДЕРЖКИ ДИНАМИЧЕСКОЙ ОТРАБОТКИ

Разработка управляющих алгоритмов и их программной реализации на БЦВМ СУ объектов РКТ представляет сложный процесс принятия и экспериментальной отработки проектных решений, реализуемых в конечном итоге в виде аттестованного ПО БЦВМ, отвечающего всем требованиям для работы в составе СУ. Этот процесс является сугубо

коллективным видом деятельности. В его реализации принимают участие коллективы специалистов различной профессиональной ориентации: автоматическое управление, навигация, системный анализ, системное и прикладное программирование, организация испытаний и др.

Средства инструментальной поддержки таких работ организованы в интегрированные технологические среды, обеспечивающие выполнение всех видов работ по проектированию, испытаниям и сопровождению в эксплуатации алгоритмического и программного обеспечения СУ. Такие среды относятся к классу специализированных CASE-сред, ориентированных на системы жесткого регламента времени (Айзенберг, Конорев, 1986).

Ниже рассматриваются основные концепции, определяющие архитектуру и реализацию CASE-среды класса систем жесткого реального времени. Предложенные концепции обеспечивают решение проблем выбора адекватных технологических схем отработки, планирования и управления работами, организации и синхронизации данных технологических процессов отработки.

КОНЦЕПЦИЯ СВЯЗАННЫХ ЦЕЛЕВЫХ КОНТУРОВ

Модель технологии отработки ПО БЦВМ может быть представлена многоконтурной схемой, включающей целевые контуры трех типов (рис. 2).

Целевой контур является абстракцией для определения типов работ (моделирование, отладка, испытания и т. д.) по отработке управляющих алгоритмов и программ БЦВМ на различных этапах общего технологического цикла разработки СУ. Концепция связанных целевых контуров создает основы для декларирования и анализа специфики технологических процессов (задач), соответст-

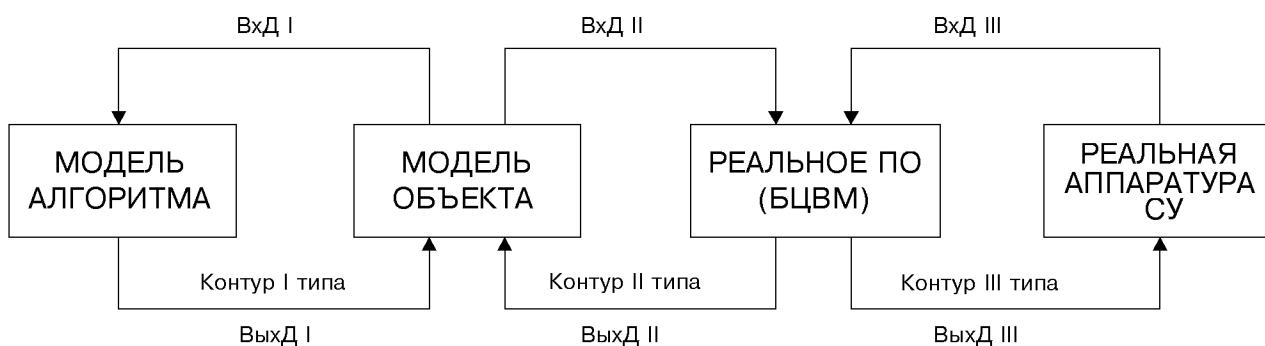


Рис. 2. Потоковая модель технологии отработки ПО БЦВМ

вующих им инструментальных средств и потоков данных в технологической CASE-среде, необходимых для отработки и аттестации ПО при оптимальном расходовании всех видов ресурсов, таких как персонал, оборудование, время.

По существу подавляющее большинство работ по отработке управляющих алгоритмов и программ БЦВМ относятся к классу задач моделирования. В предложенной классификации целевых контуров в качестве типообразующего признака выбрана характеристика инструментальной поддержки решения задач моделирования или, другими словами, тип инструментальной конфигурации контура.

Целевые контуры I типа обеспечивают решение задач математического моделирования. Основными информационными объектами таких контуров являются модели объекта управления, модели управляющих алгоритмов и модели фрагментов законов управления (исследовательские задачи), реализуемые на инструментальных ЭВМ.

Целевые контуры II типа используются для решения задач смешанного моделирования. Инструментальные средства в таких контурах представлены реальной управляющей программой, реализуемой БЦВМ, и математическими моделями объекта управления, реализуемых на инструментальных ЭВМ.

Целевые контуры III типа обеспечивают решение задач физического моделирования, при котором используется реальная аппаратура и программное обеспечение СУ, реальные агрегаты и оборудование объекта управления.

Предложенная классификация позволяет определять производные типы контуров, относящихся ко II типу (смешанное моделирование), необходимость в которых возникает в конкретных условиях в зависимости от типа объекта управления. Так, например, при работах в целевых контурах I типа могут в ряде случаев использоваться аналого-цифровые комплексы (АЦК) для реализации моделей объекта и управляющих алгоритмов. А в случае отсутствия корректных и достаточно простых для реализации математических моделей рулевых приводов систем стабилизации и регулирования в АЦК используются нагрузочные стенды, включающие реальные рулевые приводы, станции гидропитания, блоки управляемой нагрузки. Это типичный пример комбинированного использования средств математического и физического моделирования (формирование контуров II типа с использованием инструментальных средств контуров I и III типов).

Еще одним типичным примером комбинированного использования целевых контуров III и I типов являются случаи применения математических мо-

делей объекта управления для обеспечения комплексных испытаний объекта на стартовой позиции в виде «электронных» пусков с реальным полетным заданием, воспроизводящих конкретные условия предстоящего запуска объекта.

Общая спецификация задач включает
для целевых контуров I типа:

- разработку структуры управляющих алгоритмов для принятой в конкретном проекте декомпозиции СУ на подсистемы;
- исследование работы подсистем (стабилизации, выведения, регулирования и т. п.) в крайних режимах эксплуатации, включая экстремальные, с имитацией различных сочетаний допусков, погрешностей, внешних возмущений, отказов оборудования объекта и СУ и т. п.;
- выбор значений полетных заданий и настраиваемых переменных в ПЗУ БЦВМ;
- генерация эталонных результатов — контрольных примеров для работ в контурах других типов;

для целевых контуров II типа:

- отработку ПО в разомкнутых схемах с целью достижения соответствия его управляющим алгоритмам;
- автономную и комплексную отработку управляющих алгоритмов и ПО в условиях максимально приближенных к реальным с имитацией крайних режимов эксплуатации СУ;
- проведение «электронных пусков» — циклов моделирования в замкнутой схеме с реальным полетным заданием с целью верификации его, а также управляющих алгоритмов и программ для условий конкретного цикла натурных испытаний объекта;

для целевых контуров III типа:

- комплексную отработку аппаратуры, алгоритмов и ПО на автономных и комплексных стендах СУ;
- верификацию совместимости ПО и аппаратуры СУ;
- испытания на различных позициях и стенах для отработки и подготовки объекта к натурным испытаниям (контрольно-испытательные станции, технические и стартовые позиции, стенды прожига двигателей и т. п.).

Отладка и испытания с использованием целевых контуров II типа — это основной объем динамической отработки ПО БЦВМ. В контурах III типа осуществляются работы главным образом на завершающих этапах отработки СУ в целом и подготовки объекта управления к натурным испытаниям.

Обобщенная потоковая схема CASE-среды, включающая обсужденные типы целевых контуров,

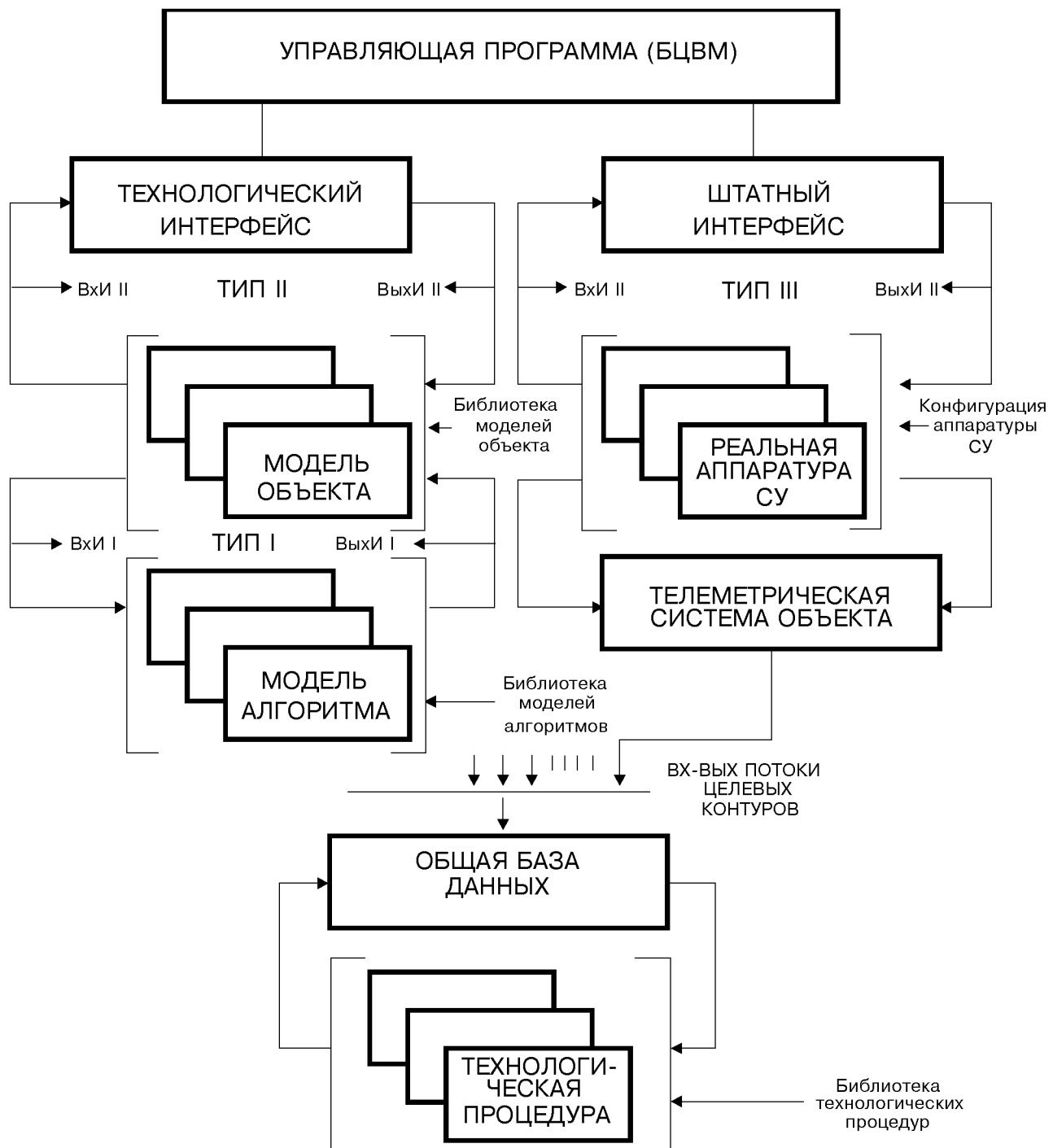


Рис. 3. Обобщенная схема потоков данных в CASE-среде при обработки ПО

приведена на рис. 3. Основные категории инструментальных средств целевых контуров — модели объекта, модели алгоритмов — представлены множествами программных компонентов библиотечного типа. Технологические процедуры, преобладающая часть которых является унифицированной для различных целевых контуров, также организованы по библиотечному принципу. Это создает основу для получения в полной мере преимуществ от использования принципа полиморфизма и позднего связывания при объектно-ориентированном подходе к реализации CASE-среды.

На схеме приведены потоки данных, имеющие смысл доставки данных от источника к потребителю:

ВхИ I, II, III — потоки входных данных для моделей алгоритмов и реальной БЦВМ;

ВыхИ I, II, III — потоки результатов (промежуточных и окончательных) реализации моделей алгоритмов или реального ПО БЦВМ.

Общая БД обеспечивает единое информационно-поисковое пространство для потоков данных всех целевых контуров.

Главной задачей CASE-среды является обеспечение эффективной методической и инструментальной поддержки коллективной разработки управляющих алгоритмов и ПО БЦВМ в рамках общей технологии создания СУ объекта управления. Критериями при этом являются обеспечение высокой производительности коллективного труда и достижение оптимальных соотношений «технологическая задача — сроки — стоимость».

Концепция связанных контуров основана на информационной совместимости целевых контуров и общей БД CASE-среды.

Информационная совместимость целевых контуров означает использование единой терминологической базы разработки СУ, совместимых интерфейсов инструментальных средств и средств междисциплинарного общения специалистов различных профессий — участников коллективной разработки СУ. Основой для реализации является использование нормативной базы (стандартов) обеспечивающей совместимость в CASE-среде языков публикации алгоритмов, языков программирования БЦВМ и языков формирования технологических спецификаций данных и управления заданиями в целевых контурах.

Общая БД CASE-среды является основой интеграции целевых контуров. Главным критерием ее организации — ориентация на процессы отработки ПО, протекающие во времени. СУ БД обеспечивает поддержку идентификации сущностей предметной области, количественное описание их, объединение

в группы технологических спецификаций, ведение и поддержание целостности.

В спецификации технологических данных входят, например, словари переменных и меток, формируемых компиляторами языков программирования, настройки каналов ввода-вывода технологического интерфейса БЦВМ, словари контрольных точек и т. п.

Основные типы информационных объектов общей БД CASE-среды представляют:

- документацию на ПО (исходные и объектные тексты, каталоги библиотек, таблицы распределения памяти, перекрестных ссылок и т. п.);
- тестовые данные (модели потоков входных, эталонных и фактических данных, статистика контрольных примеров и т. п.);
- спецификации технологических данных;
- состояния процессов отработки и т. п.

Контроль соответствия данных различных технологических операций основан на наследовании информационными объектами общей БД дат (времени) всех последовательных транзакций.

В общем случае БД целевого контура представляет собой подмножество общей БД CASE-среды.

КОНЦЕПЦИЯ СИНХРОНИЗАЦИИ ДАННЫХ ПРИ ДИНАМИЧЕСКОЙ ОТРАБОТКЕ

В реальных условиях управляющая программа в определенные моменты времени, заданные периодом квантования входной информации СУ, выдает управляющие воздействия на объект и получает от объекта по каналам обратной связи информацию об изменении его состояния.

При отработке управляющая программа должна взаимодействовать с моделью объекта аналогичным образом. Однако наряду с этим должны быть выполнены определенные технологические операции.

К таким операциям относятся:

- регистрация данных (запись фактически и/или эталонных результатов), необходимых для проверки логической структуры программы и корректности (точности) вычислений;
- включение процедур обработки результатов и передача им необходимых данных;
- включение модели объекта и передача ей управляющих параметров;
- подготовка входной информации в модели абонентов БЦВМ;
- подстановка тестовых значений в качестве текущих значений выбранных переменных для

- обеспечения автономной отладки;
- операции отладчика: отображение значений переменных во всех типах ЗУ БЦВМ, ручная установка значений, прокрутка фрагмента программы, слежение за изменением переменных и т. п.

Из-за различий временной организации управляющей программы и программной модели управляющих алгоритмов на инструментальной ЭВМ возникает проблема установления соответствия при сравнении результатов выполнения обеих программ. Решение этой проблемы основано на использовании в технологии отработки аппарата контрольных точек. Контрольная точка — это точка в адресно-временном пространстве программы, представленная в общем случае всеми параметрами текущего состояния программы. Исходным решением является назначение (декларация) контрольных точек в управляющих алгоритмах. По существу это операция определения типов и мест установки датчиков информации для проведения циклов моделирования при отработке в целевых контурах различных типов.

Контрольная точка идентифицируется уникальным именем и порядковым номером.

Руководящим замыслом обеспечения синхронизации данных (в смысле установления взаимного соответствия) при отработке является связывание контрольных точек с событиями и параметрами управляющих алгоритмов, а не с реальным временем непосредственно. Это естественным путем решает проблему установления соответствия данных при сравнении фактических и эталонных процессов и контроля логической структуры управляющей программы при отработке в контурах различных типов.

С контрольной точкой связываются технологические операции из приведенного выше перечня. Эта связь задается описанием контрольной точки, содержащим перечень операций, которые необходимо выполнить в моменты ее реализации.

Описание контрольной точки имеет следующий формат:

тип операции [имя программы] [параметры]

Тип операции указывает одну из приведенных выше технологических операций (например, включение модели объекта и передача ей управляющих параметров). Тип операции введен для того, чтобы иметь возможность программного управления блокировкой-разблокировкой операций данного типа во всех контрольных точках (без изменения их описаний) в зависимости от частной схемы испытаний (например, при испытаниях в классической замкнутой схеме процедуры обработки результатов

для разомкнутой схемы должны отключаться).

Имя программы определяет конкретную программу в пределах перечня операций данного типа.

Параметры могут задаваться явным перечислением имен, ссылкой на файл или ключом-маской, определяющим группу параметров в словаре. Все необходимые атрибуты параметров содержатся в словарях переменных программы.

Реализация контрольной точки заключается в фиксации ее прохождения программой и выполнении заданных технологических операций.

Использование общего множества контрольных точек в эталонных и фактических программах позволяет контролировать правильность выполнения программ путем сравнения соответствующих параметров в одноименных контрольных точках.

Практическое использование предложенной концепции синхронизации данных при отработке, основанной на аппарате контрольных точек, обеспечивает:

- высокую «разрешающую способность» локализации ошибок и управления технологическими процессами при отработке ПО БЦВМ. Не существует никаких принципиальных ограничений на частоту «сети» контрольных точек, кроме затрат времени на их описание и замедление процесса выполнения программы. Поэтому оптимальное назначение контрольных точек — задача не тривиальная, связанная с нахождением желаемого компромисса между затратами и эффективностью отработки ПО БЦВМ. В предельном случае может существовать только одна контрольная точка, в которой реализуются все необходимые технологические операции;
- эффективный способ синхронизации данных при использовании многоконтурных моделей технологии проектирования программно-алгоритмического обеспечения БЦВМ СУ, относящихся к классу систем жесткого реального времени. Максимальный эффект достигается для случая управляющих программ с асинхронной организацией процессов, реализуемых в их операционных системах;
- возможность контроля логической структуры и аттестации отрабатываемой управляющей программы. Зарегистрированная в процессе отработки последовательность реализаций контрольных точек позволяет представить практически реализованный маршрут на логической схеме программы. Накопление такой информации для серии испытаний дает возможность аттестовать управляющую программу по критерию покрытия маршрутов и/или ветвей ее логической граф-схемы.

КОНЦЕПЦИЯ АДАПТИВНОГО УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМИ ПРОЦЕССАМИ ДИНАМИЧЕСКОЙ ОТРАБОТКИ

Отладка и испытания представляют разные типы работ при отработке управляющей программы, сильно отличающиеся целями и условиями проведения. Если цель отладки — локализовать и устранить ошибки в условиях, когда точно известно, что они есть, то цель испытаний — подтвердить, что ошибок нет. Отладка — это эвристический пошаговый процесс, который спланировать заранее возможно только в общих чертах. Он требует постоянного участия человека. Испытания же можно проводить по намеченному плану в автоматических режимах. Реально отработка ПО систем жесткого реального времени представляет собой сложный технологический процесс, включающий трудно предсказуемые итеративные комбинации процессов отладки и испытаний.

Метод контрольных точек, описанный в предыдущем разделе, обеспечивает высокую избирательность и глубину проникновения в отрабатываемую программу, учет ее логической структуры и последовательности протекающих процессов и событий, нужную степень дифференциации отладочных и испытательных технологических процедур. Однако для полного использования потенциальных возможностей предложенного метода необходимо гибкое (адаптивное) управление. Под управлением контрольными точками понимаются способы их установки и активизации.

Возможно два способа установки контрольных точек: статический и динамический.

Статический способ заключается в предварительной установке контрольных точек на этапе проектирования программы. Используется для достижения оптимального покрытия логики программы и синхронизации вспомогательных процессов при минимальном количестве контрольных точек. Выбор размещения контрольных точек для таких операций синхронизации, как включение модели и ввод входной информации, должен производиться особенно тщательно с учетом всех динамических характеристик СУ.

Динамический способ позволяет вводить контрольные точки оперативно, в процессе отладки. При этом либо используется существующее в библиотеке описание, либо составляется новое из существующих операций и параметров.

Для придания большей гибкости, в особенности статическому способу, используется механизм активизации контрольных точек, охватывающий три уровня:

- контрольные точки или их группы;
- операции или их группы;
- параметры или их группы.

Во время испытаний реализуются только активные контрольные точки, операции и параметры. Управление активностью (включение и блокировки) контрольных точек производится внешними технологическими процедурами.

В целом же понятие адаптивного управления отработкой означает возможность приспособления путем гибкой оперативной перенастройки инструментальных средств для:

а) отработки управляющих программ СУ различных типов объектов управления, относящихся к классу систем жесткого реального времени;

б) реализации оперативно меняющихся планов и объемов отработки в условиях трудно прогнозируемых, сложно протекающих процессов отработки ПО БЦВМ в составе общей технологии разработки СУ объектов ракетно-космической техники.

Адаптивное управление работами при отработке управляющих программ является характеристикой архитектуры CASE-среды. Представляемая концепция адаптивного управления основана на использовании трех системообразующих принципов:

1. Отрабатываемая программа (ПО БЦВМ) является ведущим элементом технологических процессов отработки, формирующим в адресно-временном пространстве управляющей программы запросы-директивы на выполнение технологических операций. Способ формирования запросов-директив основывается на использовании контрольных точек и должен поддерживаться архитектурой инструментальных средств CASE-среды (наличие механизма слежения и перехвата требуемого адреса при выполнении управляющей программы на БЦВМ).

2. Внешнее, программное управление службой времени, состоянием каналов мультиплексного обмена и внутренних регистров БЦВМ. На этой основе реализуются старт-стопные режимы и динамические остановы без потерь точности реального времени при исполнении управляющей программы на БЦВМ. Процедуры управления службой времени и состоянием внутренних регистров поддерживаются архитектурой БЦВМ и являются стандартными процедурами технологической CASE-среды, выполняемыми инструментальной ЭВМ. С их помощью обеспечиваются возможности останова, фиксации состояния и результатов, коррекции текущего состояния и возобновления выполнения управляющей программы для любой точки моделируемой траектории движения объекта управления. На этом принципе достигается эффективное тестирование конфигураций ПО с древовидной логической

структурой. Трудно формализуемые процедуры отладки управляющей программы в терминах языков программирования основаны также на программной доступности ЗУ и внутренних регистров БЦВМ.

3. Гибкое формирование требуемых конфигураций инструментальных средств CASE-среды для выполнения работ в целевых контурах. Это свойство достигается на основе использования механизма установления отношений частичного порядка на множестве инструментальных средств различных связанных целевых контуров и включение его в операторский интерфейс CASE-среды. Благодаря этому обеспечивается возможности оперативного планирования отработки и реализации совмещенных режимов работы в различных целевых контурах. Методологической основой реализации концепции является объектно-ориентированный подход при проектировании и разработке CASE-среды. В результате создаются развивающиеся библиотеки различных классов объектов с использованием принципа наследования свойств в пределах класса, стандартные механизмы динамического взаимодействия инструментальных средств (объектов) на основе посылки и приема сообщений и средства сборки требуемых конфигураций инструментальных средств на основе принципов полиморфизма, позднего связывания и управляющего монитора.

АРХИТЕКТУРА ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ ДИНАМИЧЕСКОЙ ОТРАБОТКИ В CASE-СРЕДЕ.

Структурная схема подсистемы комплексной динамической отработки ПО БЦВМ в CASE-среде представлена на рис. 4. Образование требуемых для конкретной СУ конфигураций инструментальных средств осуществляется путем реализации запроса на сборку из библиотечных компонентов и настройки их на характеристики конкретной СУ.

Типовая спецификация необходимых аппаратных средств инструментальной среды в общем случае включает модули:

- формирования и выдачи по программе инструментальной ЭВМ команд управления БЦВМ;
- последовательный порт ввода-вывода для организации двухстороннего обмена функционально маркированными запросами-директивами между отрабатываемой программой БЦВМ и инструментальной ЭВМ. Организация доступа БЦВМ ко всем абонентам инструментальной ЭВМ функционально эквивалентна оснащению БЦВМ всем периферийным оборудованием инструментальной ЭВМ (videотерминал, НЖМД,

принтер и т. п.);

- порт обмена между инструментальной ЭВМ и БЦВМ параллельными кодами. Данный порт поддерживает выполнение директив-запросов настройки инструментальной среды на отработку конкретной программы БЦВМ, директив обмена информацией между отлаживаемой программой и инструментальной ЭВМ, организацию диалогового режима взаимодействия с отрабатываемой программой и т. п.;
- технологических прерываний отлаживаемой программы по заданным временным и программным координатам (динамический останов БЦВМ) и, как следствие, возможность априорного и оперативного планирования процесса отладки программ БЦВМ путем расстановки контрольных точек (в том числе и «бесконтактным» методом, без ввода в отрабатываемую программу технологических вставок);
- блокировки службы реального времени БЦВМ и мультиплексных каналов обмена (КМО) с абонентами СУ при динамических остановах отрабатываемой программы.

В частных случаях перечисленные модули могут быть элементами архитектуры инструментальных или встроенных ЭВМ. В состав базы данных в общем случае входят:

- база данных проекта СУ;
- база данных программных компонент CASE-среды.

Опыт АО ХАРТРОН показывает, что для генерации в CASE-среде конфигураций инструментальных комплексов динамической отработки ПО БЦВМ, отвечающих требованиям СУ различных типов объектов управления, достаточной является следующая спецификация классов базовых программных компонентов:

- драйверы обмена между БЦВМ и инструментальной ЭВМ по параллельному и последовательному портам;
- динамические настройки драйверов обмена на реализацию конкретной директивы-запроса;
- статические настройки на конкретный проект СУ;
- монитор реализации директив-запросов.

Интерфейс пользователя CASE-среды обеспечивается с помощью меню двух классов:

- меню одиночных директив-запросов;
- меню пакетов директив-запросов.

Типовая спецификация директив-запросов состоит из следующих классов объектов:

- загрузка отлаживаемой программы из базы данных в ЗУ БЦВМ;
- загрузка настроек алгоритмов СУ и настроек

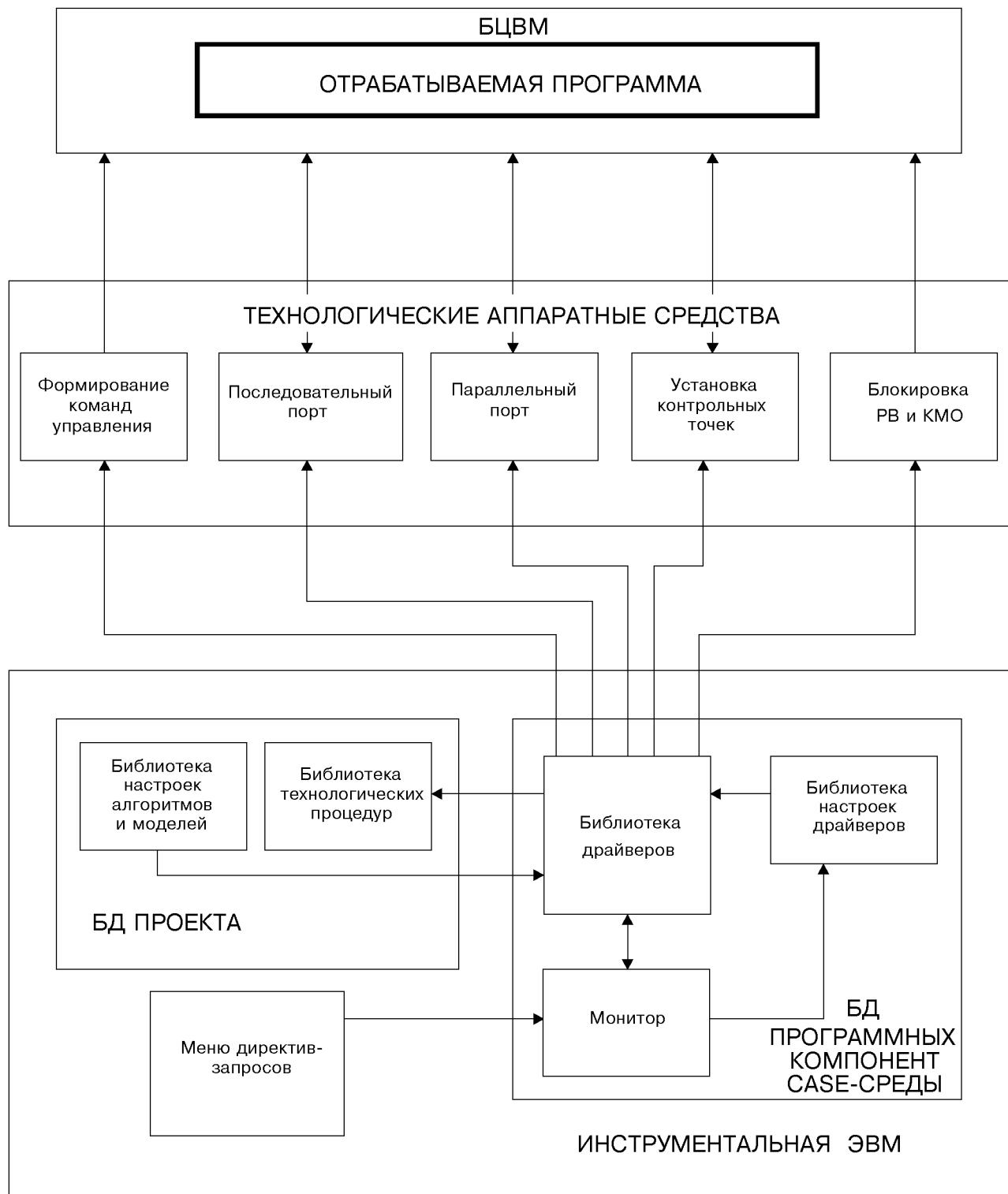


Рис. 4. Архитектура инструментальных средств динамической отработки ПО БЦВМ

- математической модели объекта управления из БД в ПЗУ БЦВМ;
- установка исходного состояния ЗУ БЦВМ, необходимого для отлаживаемой программы;
- установка исходного состояния моделей бортовых и наземных абонентов СУ;
- настройка параметров цифровых моделей объектов управления;
- формирование списка временных и программных координат динамических остановов отлаживаемой программы;
- корректировка параметров в ЗУ БЦВМ;
- формирование временных координат и перечня параметров, записываемых в БД и выдаваемых на видеотерминал или принтер;
- формирование списка временных и программных координат участков отлаживаемой программы, на которых включается режим трассировки программных модулей;
- обработка тактовой метки (метка конца цикла решения алгоритмов управления);
- включение цифровой модели объекта управления;
- прием и запись в БД технологических кадров, содержащих результаты работы отлаживаемой программы;
- формирование и запись в БД трассы (маршрута) заданного участка программы в заданном интервале времени;
- прием и запись в БД координат переполнения разрядной сетки БЦВМ.

Объектно-ориентированный подход обеспечивает специализацию и необходимые модернизации перечисленных классов объектов путем оперативного создания подклассов и развития библиотек CASE-среды (драйверы, технологические процедуры, директивы).

ЗАКЛЮЧЕНИЕ

1. Технология отработки и аттестации ПО БЦВМ СУ ракетно-технической тематики является высокоспециализированной и существенно отличается от технологии ПО других типов, например ПО АСУТП промышленного профиля.

Основные причины специализации:

- уникальность задач СУ конкретного типа ракеты или космического аппарата;
- невозможность проведения отработки ПО с реальным объектом, принципиальное отсутствие возможности проведения тестов для отработки ПО в основных режимах работы ракеты или космического аппарата;

- прямая зависимость логики, точности, динамики СУ в целом от характеристик ПО БЦВМ (ПО БЦВМ функционирует непосредственно в контурах управления необратимыми операциями и движением объекта) и высокая (а в ряде случаев катастрофически высокая) цена ошибки в ПО, следствием чего является необходимость решения для динамической отработки ПО проблем адекватной имитации условий, максимально приближенных к реальным условиям основных (полетных) режимов ракеты или космического аппарата.

2. Изложенные в статье концепции технологии динамической отработки и аттестации ПО БЦВМ являются обобщением опыта работ АО ХАРТРОН, начиная с 1964 г., и представляют важную составную часть прикладной теории программирования встроенных ЭВМ систем жесткого реального времени.

Корректность и практическая эффективность концепций подтверждена положительными результатами динамической отработки и аттестации ПО БЦВМ при проведении свыше 300 пусков (циклов натурных испытаний) ракет и космических аппаратов различных типов.

3. На различных этапах работ существенный вклад в проведение теоретических и экспериментальных исследований, создание основ прикладной теории и практическую реализацию технологии динамической отработки и аттестации ПО БЦВМ внесли следующие специалисты АО ХАРТРОН:

Бондарев А. И., Батаев В. А., Воронченко В. Г., Герасименко В. И., Гристан А. С., Гончар А. С., Дудник Е. И., Демидов Е. Н., Корума С. С., Крупский В. С., Красногоров В. С., Калногуз А. Н., Кириченко А. Ф., Лысенко А. А., Макаренко А. Я., Рыжавская Н. Г., Романенко В. Н., Симагин В. Г., Стадник В. Д., Сухоребый В. Г., Стрелец А. А., Филиппова К. А., Чумаченко В. И., Черкашин В. Д., Чернышов М. А., Юрченко Л. Д. и многие другие.

Всем им авторы выражают глубокую признательность.

Айзенберг Я. Е., Конорев Б. М., Щербаченко В. Т. и др.
Комплекс автономного управления ракеты-носителя «Энергия». Теоретические принципы построения, управляющие алгоритмы и программы, разработка, отработка, натурные испытания. Цикл работ. — Харьков, Хартрон, 1985—1990 г.г.

Айзенберг Я. Е., Бек А. В., Лысенко А. А. и др. Система управления транспортного корабля снабжения и функционально-грузовых орбитальных модулей «Квант», «Кристалл», «Спектр», «Природа». Теоретические принципы построения, управляющие алгоритмы и программы, разработ-

- ка, отработка, натурные испытания. Цикл работ. — Харьков, Хартрон, 1966—1996 гг.
- Айзенберг Я. Е., Златкин Ю. М., Каменев В. П. и др. Система управления семейства межконтинентальных баллистических ракет. Теоретические принципы построения, алгоритмы управления и контроля, ПО БЦВМ, разработка, отработка, натурные испытания. Цикл работ. — Харьков: Хартрон, 1964—1991 гг.
- Айзенберг Я. Е., Конорев Б. М. Организация имитационного моделирования в автоматизированных системах производства программ реального времени // УСиМ.—1982.—№ 4.—С. 83—87.
- Айзенберг Я. Е., Конорев Б. М. Концепция технологической среды производства программного обеспечения встроенных ЦВМ систем реального времени // Технология программирования: Докл. II Всесоюз. конф. — Киев, 1986.—С. 13—19.

DYNAMIC TESTING OF ON-BOARD EMBEDDED COMPUTER SOFTWARE OF MISSILE AND SPACECRAFT CONTROL SYSTEM

Ya. E. Aisenberg, A. V. Bek, Yu. M. Zlatkin,
V. P. Kamenev, B. M. Konorev, and V. T. Shcherbachenko

For a wide range of missiles and spacecraft the on-board embedded computer (BEC) software as an element of a guidance and control system (G&CS) determines to a considerable degree the logic, dynamics and accuracy of functioning of the object under control,

being an implementation of control algorithms. Real time interaction of BEC software with real equipment directly within the control loop of complex irreversible process in main performance modes predetermines ultimately hard requirements as to the quality of its testing and enormous labour consumption. As a result there is a crucial dependence of the quality, development time, and cost of the control system as a whole on BEC software. Using the adequate BEC programming technology is the principal way of solving this problem. Goals and significance of dynamic testing in the form of "electronic launches" are determined on a qualitative level within the context of general technology of creating G&CS. Methodology of dynamic BEC software testing which is considered to be the most important part of application theory of programming BEC for missile and spacecraft G&CS, which belong to the hard real-time systems, is described informally. Block diagrams of BEC software testing are discussed in terms of data flow model, and principal concepts of their CASE tool support realization, i. e., concepts of coupled target loops, data synchronization and adaptive process control have been stated. Materials of the article present a generalization of Hartron experience starting from the 60s in the field of creating software for G&CS BEC in the main areas of missile and spacecraft industry: heavy rocket carrier "Energia", a family of functional modules "Kvant", "Krystall", "Spektr", "Pryroda", orbital manned stations, a family of intercontinental ballistic missiles with separating heads including missiles of SS-18, SS-19 types, transportation vehicles and power-supply functional and transport module in the American-Russian project of the international space station "Alfa", etc. Correctness and practical efficiency of the concepts have been confirmed by the results of BEC software testing in support of above 300 launches of missile and space vehicles of various types. The article is aimed at experts engaged in investigations, technologies, and practical management in the programming of BEC for hard real-time systems.