

шить все мыслимые и немыслимые ошибки – вреда или убытков это не принесет) и начать процесс бурения, рассчитывая для него оптимальные параметры.

Предлагая обучаемым интерактивный режим обучения и активно взаимодействуя с ними, система дает возможность:

- изучить методику расчета оптимальных параметров бурения;
- получить необходимые навыки для осуществления процесса бурения скважин на оптимальном режиме;
- освоить технологию отработки долот с целью достижения максимальных ТЭП в процессе проводки скважин при данном техническом оснащении буровой установки.

И все это – не отходя от рабочего стола в кабинете.

Тренажер моделирует процессы оптимизации бурения скважин и обеспечивает:

- индикацию начальных и текущих параметров бурения на экране дисплея;

- динамику ТП;
- выполнение расчетов оптимальных параметров бурения.

В процессе обучения пользователя тренажер-имитатор позволяет преподавателю производить: выбор режима обучения и задачи; выбор и ввод управляемых воздействий; анализ и интегральную оценку действий обучаемого; вывод протокола решения задач на экран дисплея и принтер; многократный тренинг для закрепления знаний и навыков; накопление информации о ходе обучения.

Следует отметить, что при разработке тренажера-имитатора "Оптимизация процесса бурения скважин" были учтены учебные программы, согласованные с Госгортехнадзором России, а сам он рекомендован для обучения и повышения квалификации специалистов нефтяной и газовой промышленности Управлением Северо-Западного Округа Госгортехнадзора России.

*Ройтман Евсей Маркович – руководитель направления разработки тренажерных систем,
Саямов Аркадий Эдуардович – руководитель отдела разработок НПК "Генезис знаний".*

Контактные телефоны: (8462) 34-87-57, 17-30-69.

[Http:// www.kg.ru](http://www.kg.ru)

ПРОГРАММНЫЙ КОМПЛЕКС "ЭНИКАД" ДЛЯ СОЗДАНИЯ МОДЕЛИРУЮЩИХ КОМПЛЕКСОВ СЛОЖНЫХ ТЕХНОЛОГИЧЕСКИХ ОБЪЕКТОВ

В.А. Чернаков, С.Б. Выговский, М.А. Осадчий, К.П. Кориковский, Ю.В. Краюшкин
(Компания ЭНИКО ТСО, ТЕХНОПАРК МИФИ)

Проанализированы факторы, влияющие на развитие компьютерных моделей сложных технологических объектов (СТОУ). Описаны история создания, архитектура, особенности и принципы использования комплекса ЭНИКАД, предназначенного для разработки и эксплуатации компьютерных тренажеров СТОУ.

К сложным технологическим объектам управления (СТОУ) в данной статье относятся объекты управления с числом контролируемых переменных и управляющих органов от нескольких сотен до десятков тысяч. Типичными примерами подобных объектов являются атомные и тепловые электростанции, ядерные энергетические установки различного назначения, химические и нефтехимические производства, а также крупные силовые установки, например, судовые. Эти объекты отличаются высокой "связностью" происходящих в них физических процессов как между собой, так и с работой АСУТП, являющейся необходимым атрибутом таких объектов.

Сегодня проектирование, эксплуатация и оптимизация ТП в СТОУ уже невозможны без использования расчетных моделей. История развития компьютерных моделей СТОУ в различных областях насчитывает уже несколько десятилетий, но в России особенно бурно эта отрасль знаний стала развиваться в последнее время.

Сформулируем основные факторы, повлиявшие на развитие компьютерного моделирования.

1. Быстрый рост вычислительной мощности ПК привел к тому, что сегодня она уже достаточна для расчета в режиме РВ комплексных моделей даже таких сложных объектов, как атомные электростанции. Высокие графические возможности современных ПЭВМ позволяют разрабатывать высокоразвитые графические интерфейсы, делающие "легким" общение с моделями, имеющими тысячи контролируемых параметров и органов управления, даже для технологов, не имеющих специальной компьютерной подготовки. Оба эти обстоятельства фактически "переводят" комплексные модели СТОУ из разряда "уникальных инструментов", доступных лишь небольшому числу специалистов-расчетчиков в профильных НИИ, в разряд массового высокоэффективного инструмента для решения различных производственных задач легко доступного как любым специалистам НИИ, так и технологам конкретных объектов.

2. Высокая степень износа и моральное старение АСУТП крупных российских предприятий, построенных в советские времена, приводят к необходимости модернизации АСУТП. На смену средствам локальной

автоматики на многих предприятиях приходят современные АСУТП, построенные на аппаратных и программных средствах ведущих западных фирм таких, как Siemens, Honeywell и др. Вместе с тем, внедрение современных западных АСУТП предполагает более строгую процедуру их проектирования: все алгоритмы защит, блокировок, сигнализации, параметры настройки регуляторов должны быть полностью верифицированы заранее до ввода их в память цифровых модулей АСУТП. Понятно, что полностью такую верификацию можно осуществить только на комплексной модели СТОУ.

3. Все более жесткими становятся требования надзорных органов к подготовке персонала СТОУ, особенно взрывоопасных и ядерно-опасных. Для таких СТОУ уже сегодня требуется полноценная тренажерная подготовка персонала во всех режимах, включая сложные аварийные. В основе тренажеров, на которых может вестись такая подготовка, должна лежать комплексная всережимная модель, адекватная реальному объекту.

Следует подчеркнуть, что современное состояние вычислительной техники и методов моделирования дает возможность создавать многофункциональные моделирующие комплексы на основе стандартных ПЭВМ с процессорами Intel, позволяющие решать все вышеперечисленные задачи даже для наиболее сложных объектов управления таких, как атомные электростанции. Это делает экономически целесообразным разработку многофункциональных моделирующих комплексов для СТОУ в различных отраслях промышленности.

Сегодня уровни развития моделирующих комплексов РВ для СТОУ сильно отличаются в зависимости от отрасли. В одних отраслях, например, в атомной энергетике, накоплен значительный опыт разработки таких комплексов, в других отраслях движение по этому пути лишь начинается. Для них был бы крайне полезен опыт атомной промышленности, поскольку стоимость разработки современного всережимного моделирующего комплекса РВ для СТОУ достаточно высока.

Создание современного моделирующего комплекса для СТОУ невозможно без специализированной оболочки, охватывающей все этапы процесса его разработки. Такие оболочки, которые в значительной мере позволяют автоматизировать процесс разработки моделирующего комплекса СТОУ, имеются у всех ведущих зарубежных тренажеростроительных фирм таких, например, как STN ATLAS ELEKTRONIK, CORYS, CAE. Аналогичная оболочка, создававшаяся в течение 11 лет и получившая название ЭНИКАД, была разработана российской фирмой ЭНИКО ТСО, входящей в Технопарк МИФИ.

Комплекс ЭНИКАД предназначен для разработки и эксплуатации компьютерных моделирующих систем со специализированным графическим интерфейсом. Класс этих систем весьма широк и включает тренажеры, анализаторы, системы-советчики, визуализаторы архивных или текущих данных различных технологических объектов, включая наиболее сложные такие, как атомные и тепловые электростанции, химические и нефтехимические производства.

Откуда появился ЭНИКАД? В начале 90-х годов в Россию хлынули ПК. Сразу стали очевидны выгоды работы с цветными графическими изображениями, удобство создания программ и небольшая стоимость ПК перевесили все минусы. Так в ЭНИКО ТСО родилось новое направление работ – компьютерное моделирование в сочетании с широким использованием графического интерфейса.

Пробором комплекса ЭНИКАД стала система, работающая под MS DOS, которую назвали ГИМ – Графический Интерфейс с Моделью. В ГИМ была заложена плодотворная идея разделения моделирующей и интерфейсной частей программы. Связующим звеном служили сами переменные модели, расположенные в COMMON-областях. Графическая часть строилась из отдельных элементов, ассоциированных с этими переменными. После каждого шага счета модели значения переменных передавались графическим элементам, что вызывало их перерисовку. Действия пользователя, в свою очередь, изменяли значения других переменных, тем самым, становясь доступными модели. Синхронизация переменных производилась исполняющей системой ГИМ автоматически на основании специального описания.

Выбранный подход переводил ГИМ в разряд инструментальных средств. Независимость моделирующей и интерфейсной части позволяли легко модернизировать модель и экранные форматы, объединяя их только при запуске системы. На основе ГИМ в середине 90-х годов был создан ряд тренажеров для предприятий атомной энергетики (Курская АЭС, Южноукраинская АЭС и др. – см. сайт фирмы <http://www.eniko.ru>).

Однако вскоре проявились недостатки реализации системы ГИМ. Препятствием на пути создания масштабных проектов стала ОС DOS, предоставляющая слишком мало ресурсов программе. Появление и широкое распространение в 1995 г. 32-разрядной системы Windows стало сигналом к созданию нового программного комплекса, который получил название ЭНИКАД. При разработке учитывались функциональные особенности аналогичных систем ведущих тренажеростроительных фирм STN ATLAS ELEKTRONIK (Германия), CORYS (Франция), GPI (США), с которыми сотрудникам ЭНИКО ТСО довелось детально познакомиться в процессе совместных работ.

Какие задачи ставили перед собой разработчики ЭНИКАД? Сохраняя архитектурные достоинства ГИМ, было решено развивать систему в следующих направлениях:

- создание единой среды разработки-выполнения для удобства реализации и сопровождения прикладных систем;
- интеграция в систему полнофункционального графического редактора для создания графического интерфейса;
- поддержка и автоматизация создания моделирующего кода;
- создание масштабируемой исполняющей системы, способной функционировать на многопроцессорных и сетевых комплексах;
- поддержка эксплуатации прикладных систем.

Основным компонентом ЭНИКАД является программа GIW (Graphical Interface Workshop). Программа реализована по модульному принципу — к ядру GIW могут добавляться специализированные подсистемы (в виде DLL), обеспечивающие требуемую функциональность создаваемой системы. Сборка комплекса производится на этапе загрузки на основании файла конфигурации.

Попробуем разобраться в архитектуре GIW (рис. 1). Базовыми понятиями являются модель и экранные форматы.

Термин "модель" используется в контексте GIW в качестве синонима для источника данных, поступающих в РМВ. Например, моделью может служить программа доступа к БД или интерфейсная программа с оборудованием — источником данных. Модель создается из программного кода, который пишется вручную на языке Фортран или Си, либо получается автоматически с помощью одной из интегрированных в ЭНИКАД систем автоматического проектирования (САПР).

Форматы, в свою очередь, представляют собой рисунки в специальном векторном представлении, изображаемые графическим модулем в окнах Windows. Форматы могут быть разных типов. Интерфейсные форматы являются средством управления и представления данных модели, внешний вид и функциональность таких форматов полностью в руках разработчика. Отладочные форматы в виде таблиц или графиков также обеспечивают взаимодействие с моделью. Наконец, форматы САПР, нарисованные по определенным правилам, являются источником получения программного кода.

Строительным материалом для форматов всех типов служат, так называемые, графические элементы — специальные объекты со специфическим визуальным представлением и функциональностью.

В настоящий момент разработано большое число различных элементов, которые можно поделить на три класса.

Статические элементы включают базовые графические примитивы. Множество статических элементов простирается от простейших (отрезков, прямоугольник, эллипс, текст и т.д.) до весьма сложных (фигуры на основе кривых Безье, градиентные заливки, видеофрагменты). Статические элементы предназначены в основном для создания фонового изображения интерфейсных форматов.

Динамические элементы — это разнообразные изображения, изменяющие свой вид в зависимости от значений переменных модели или, наоборот, изменяющие переменные модели при определенных действиях пользователя. Динамические элементы бывают универсальными (типа кнопок или графиков) или специализированными (изображения арматуры конкретной технологической установки). Динамические элементы являются основой для организации интерфейса с моделью и всегда ассоциированы с одной или несколькими переменными модели.

Элементы САПР представляют собой базу для создания схем различных типов. Характерной особенностью таких элементов является способность порождать фрагменты моделирующего программного кода. Для каждого типа САПР существует свой набор первичных элементов. Для представления типовых узлов оборудования из первичных элементов могут быть созданы макроэлементы, используемые в дальнейшем наравне с остальными элементами.

Графические элементы оформляются в виде подгружаемых библиотек. Идея использования библиотек очень плодотворна. С одной стороны — это прекрасный способ наращивать и подстраивать элементную базу под нужды конкретной разработки. С другой стороны, ненужные для проекта библиотеки можно исключать, разгружая систему от лишних объектов.

Все аспекты функционирования форматов с размещенными на них элементами находятся под управлением графического модуля GIW. По сути, этот модуль представляет собой универсальный графический редактор с развитым набором функций. Объектно-ориентированная природа элементов позволяет легко модернизировать форматы. Каждый элемент имеет, так называемые, свойства. Изменяя свойства элемента, разработчик настраивает его под свои нужды. Важно подчеркнуть, что любые процедуры редактирования или настройки элементов выполняются единообразно для любого элемента в системе.

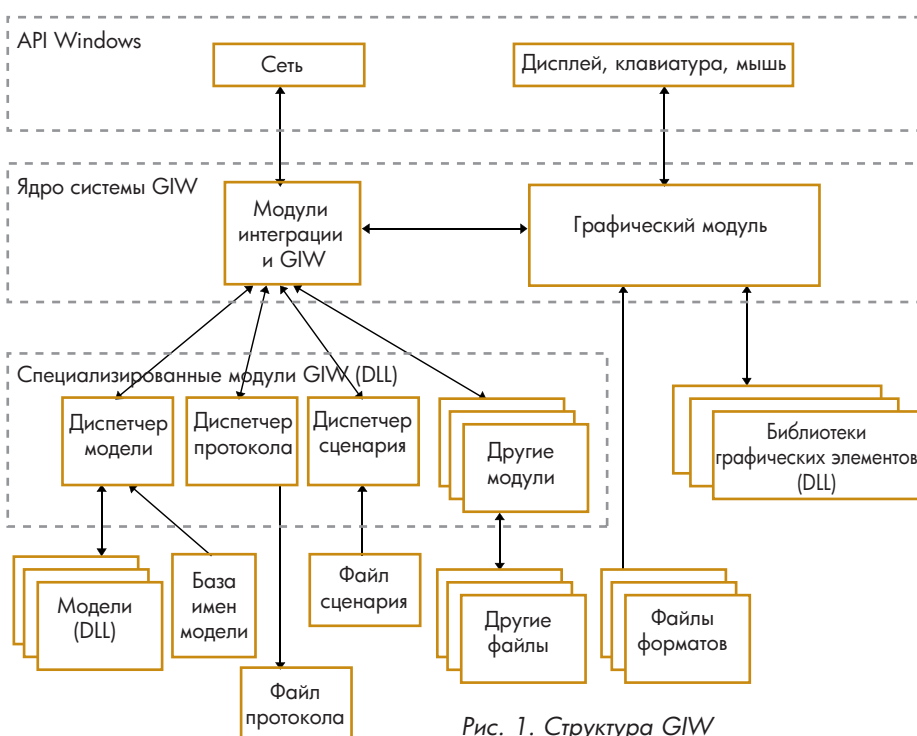


Рис. 1. Структура GIW

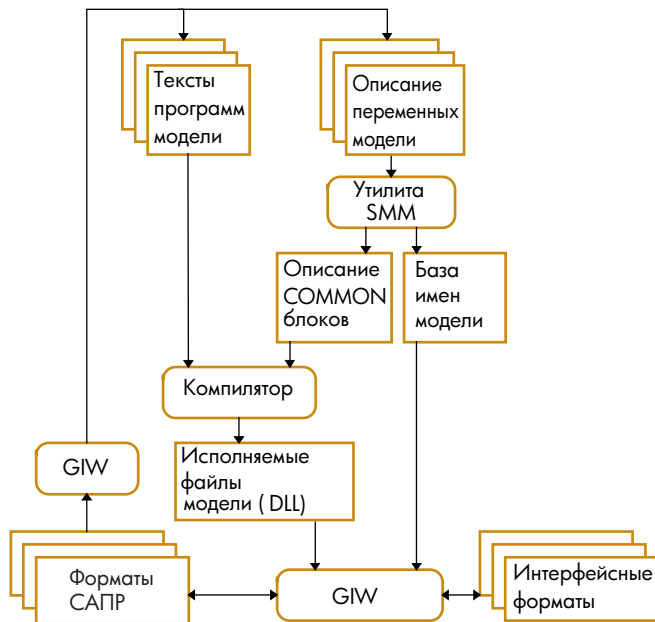


Рис. 2

Посмотрим, как реализован поток информации в системе (рис. 2). Основным способом обмена данными между моделью и остальными частями системы является организация совместного доступа к COMMON-областям модели. Для глобальной идентификации данных используются синонимы программных идентификаторов, так называемые технологические имена, обычно хорошо известные специалистам-технологам моделируемого объекта. При этом осуществляется разделение пространства имен различных программных объектов при обеспечении пространства глобальных имен, общих для всего проекта. Данный подход позволяет:

- эффективно декомпозировать модель в рамках проекта и программировать отдельные модули (объекты) независимо один от другого;
- на базе согласованного интерфейса вести параллельную разработку программных объектов проекта, в том числе разными коллективами разработчиков;
- использовать русские глобальные имена переменных проекта, что дает возможность моделировать все без исключения отечественные технологические объекты.

Поддержка связей моделей средствами GIW обеспечивает корректную передачу данных между различными объектами проекта в однопроцессорной, мультипроцессорной и сетевой вычислительных средах. В то же время, исполняющая система GIW гарантирует разделение и сохранение данных отдельных программных объектов, не участвующих в связях, независимо от используемых совпадающих программных имен.

Рассмотрим другой аспект функционирования GIW – поддержку диспетчеризации моделей РВ. Исполняющая система GIW обеспечивает вызов моделей в соответствии с фиксированной сеткой вызовов, привязанной к РВ. В пределах этой сетки для каждого про-

граммного объекта могут быть установлены шаг и фаза вызова, а также очередность вызовов моделей, у которых совпадают шаг вызова и фаза. Это обеспечивает правильный старт комплексной модели, а также позволяет строить различные численные схемы средствами исполняющей системы, эффективно распределять процессорное время между различными программными объектами с сильно различающимися динамическими свойствами. Минимальный шаг не ограничен и определяется только потребностями моделирования и вычислительной мощностью процессора. Диспетчер моделей реализует также важную сервисную функцию – замедление или ускорение темпа моделирования.

Для обеспечения работы модели сложных ТП в РВ имеется возможность распараллеливания процесса вычислений с использованием многопроцессорных вычислительных структур. Исполняющая система GIW полностью берет на себя решение вопросов диспетчеризации и синхронизации моделей, а также обмена данными.

Другой способ повышения производительности – организация сетевой вычислительной среды. При моделировании в РВ сложных систем, в том числе требующих многотерминального графического интерфейса, GIW позволяет размещать модели отдельных систем на отдельных машинах, связанных по сети. При этом синхронизация моделей, связь с базой имен моделей, экраным интерфейсом и средствами отладки осуществляется средствами GIW.

Для функционирования прикладных систем обычно возникает потребность в ряде сервисных функций. Эти функции реализуются в GIW специализированными подгружаемыми модулями.

Диспетчер протокола реализует возможность сохранения параметров процесса и управляющих воздействий в файле протокола. Наличие протокола позволяет на количественном уровне исследовать поведение модели, оценивать динамическую реакцию на отдельные события, сравнивать поведение модели в различных условиях работы, накапливать базу протоколов для последующего анализа. По графику протокола в целом или по любой его части можно получить распечатку для документирования результатов численного эксперимента.

Диспетчер сценариев обеспечивает возможность задания управляющих воздействий на модель из сценария, написанного заранее на простом мета-языке. Система используется для создания тестовых или учебных задач. Например, сценарий позволяет унифицировать условия проведения серии тестов для сравнения результатов различных настроек параметров моделей.

Подсистема работы с состояниями модели реализует следующие функции:

- сохранение полного состояния модели процесса или любой его части по выбору;
- загрузку нового состояния модели или любой его части (например, констант настройки отдельного программного модуля);
- откат модели на запомненное предыдущее состояние, число возможных откатов и шаг по времени можно регулировать.

Закончив обзор основных подсистем GIW, рассмотрим теперь, как происходит создание прикладной системы.

GIW является интегрированной средой и обеспечивает процесс разработки разнообразных технологических систем на всех стадиях: создания моделей и графических форматов, автономной отладки, сборки, комплексной отладки и эксплуатации. Как правило, процесс разработки прикладной системы, за исключением создания исполняемых файлов модели, производится непосредственно в среде GIW. Для систематизации рабочих файлов различных типов используется встроенный менеджер проекта.

Процесс разработки прикладной системы в среде GIW: с использованием менеджера проекта пользователь работает только с форматами САПР и интерфейсными форматами.

В GIW предусмотрена декомпозиция сложных моделей на логически законченные подсистемы, реализуемые в виде отдельных загружаемых модулей. Такой подход обеспечивает возможность автономной разработки отдельных частей модели с последующей их интеграцией на заключительных этапах проектирования. Модели в GIW могут строиться как при помощи встроенных в него средств автоматизации (САПР моделей автоматике, теплогидравлических систем и электрических систем), так и вручную. Познакомимся с особенностями САПР разных типов.

Модель сложной технологической системы, как правило, включает модели физических процессов, происходящих в системе, а также модели технологической автоматике, защит, блокировок и управления, т. е. модели логико-динамических систем.

САПР систем автоматике предназначен для создания моделей логико-динамических систем. Структура такой системы включает набор динамических звеньев и функциональных преобразователей аналоговых и логических сигналов, соединенных друг с другом направленными связями. Расчет параметров состояния каждого звена схемы производится последовательно, а порядок обхода элементов схемы определяется на основе анализа направленного графа этой схемы, с учетом возможных обратных связей. Для выявленных контуров обратных связей производится коррекция запаздывания на расчетный шаг.

В основе представления исходной информации о структуре системы, ее элементах и параметрах лежит графическое представление эквивалентной технологической схемы системы (см. иллюстрации на второй полосе обложки).

Элементы схемы реализуют все возможные динамические, статические или специализированные функциональные элементы системы. Такие элементы поставляются в виде библиотеки стандартных элементов в составе САПР автоматике, а также могут изготавливаться самим пользователем на базе уже существующих элементов. Для изготовления специализированных элементов схем используется та же самая графическая среда, что и для разработки самих схем системы. Программный код элемента может быть получен как при обработке соответствующей

схемы элемента, так и написан вручную с использованием всех средств и возможностей программирования в стандарте FORTRAN-90. Это дает возможность наряду с большой наглядностью и простотой схемы использовать для обработки сигнала сложные программные алгоритмы.

Мнемоника элементов для вставки в схемы САПР автоматике создается одновременно с описанием кода элемента, его данных и точек подключения, и соответствует стандартным изображениям соответствующих логических, аналоговых или смешанных элементов.

При таком подходе к изготовлению элементной базы схем САПР появляется возможность моделирования схем устройств технологической автоматике и логики методом прямой аналогии с отражением на ней всех реальных элементов схемы устройства и связей между ними. Большая по объему схема САПР автоматике может быть разбита на произвольное число отдельных листов схемы с сохранением связей между ними. Такое множество листов схемы системы обрабатывается как единая схема.

Результатом компиляции схемы САПР автоматике является программный код этой схемы, а также файлы объявления переменных, используемых этой схемой, и начального состояния переменных схемы САПР. Начальные значения переменных состояния элементов схемы САПР задаются в интерактивном режиме при рисовании схемы, и позволяют "стартовать" такую схему как автономно, так и в составе комплексной модели без ложных импульсных сигналов или переходных процессов, не обусловленных внешними сигналами. Также в интерактивном режиме при проектировании схемы САПР задаются все константы – параметры настройки элементов схемы.

САПР систем автоматике позволяет производить автономную предварительную отладку технологической системы без включения ее в полный проект. Для этого на основе схемы САПР автоматике собирается отдельная модель, которая может быть запущена в отсутствие других моделей под управлением GIW.

Среда разработки позволяет контролировать сигнал в любой точке схемы САПР автоматике в РВ, изменять параметры настройки элементов.

Для связи с другими системами в рамках комплексной модели используются элементы типа "разъем". В случае автономной работы полученной модели системы они выступают как источники граничных условий для модели системы.

Каждый лист схемы моделируемой системы оформляется в соответствии с требованиями, предъявляемыми к аналогичной конструкторской документации. Распечатка таких схем системы является полным отчетным документом по модели этой технологической системы и содержит всю информацию о структуре связей элементов схемы, параметрах их настройки, внешних связях системы и необходимых параметрах начального состояния системы.

Средствами САПР автоматике создан ряд функциональных модулей управляющих систем автоматике, включающий основные элементы ряда УКТС для уп-

правления основными приводами, а также ряд элементов комплексов КАСКАД, предназначенных для построения схем основных регуляторов энергоблоков с реакторами ВВЭР и РБМК. На основе наработанной общей и специализированной элементной базы производится детальное моделирование управляющих систем энергоблоков ВВЭР и РБМК для создания моделирующих комплексов класса аналитических тренажеров энергоблоков. В настоящее время создается библиотека элементов для системы TELEPERM-ME (разработки Siemens), известной в России под названием ТПТС-51.

САПР моделей теплогидравлических систем предназначен для создания динамических моделей теплогидравлических сетей и разнообразного оборудования в их составе. Он позволяет проектировать модели с использованием набора стандартных (базовых) элементов теплогидравлики таких, как объемы воды и пара, трубы, ступени турбин, насосы, теплообменные аппараты, обратные клапана, запорная и регулирующая арматура, датчики основных параметров КИП и т. п., а также специализированных (составных) элементов.

При математическом моделировании теплогидравлических систем используется неявная нелинейная схема с внутренними итерациями. На каждой итерации решается система уравнений относительно приращения переменных состояния на данной итерации. Такой подход позволяет существенно уменьшить влияние погрешностей вычислений на результат.

Элементы САПР разрабатываются с помощью редактора элементов САПР, входящего в состав GIW. Разработка новых элементов доступна конечному пользователю. Мнемоника элементов схем САПР в основном соответствует стандартным технологическим схемам теплогидравлических систем. Большая по объему теплогидравлическая схема может быть разбита на произвольное число отдельных листов с сохранением связей между ними. Фрагмент теплогидравлической схемы представлен на второй полосе обложки. Результатом компиляции схемы САПР теплогидравлики является программный код на языке Фортран.

Разумеется, и в этом случае среда разработки предоставляет полный набор возможностей по настройке, отладке и документации моделей.

САПР моделей электрических систем построен аналогично САПР теплогидравлики. Он позволяет проектировать модели основного и вспомогательного оборудования электростанций, а также сетевые структуры энергосистем. Базовыми объектами являются модели таких устройств, как генератор, электродвигатели различных типов, трансформаторное оборудование, кабельные и воздушные линии электропередачи, сеть бесконечной мощности, измерительные приборы в точках подключения. Моделируется влияние коммутаций на топологию сети.

Важным этапом в ходе разработки любой модели является процесс отладки. Среда разработки предоставляет возможность отладки как отдельных модулей, так и полной модели. Отладка может производиться в РВ и в пошаговом режиме. Предоставлена возможность получения информации о последовательности

вызовов моделей, результатах таймирования отдельных моделей и модели системы в целом. Накопление статистики о времени работы отдельных моделей системы, выполняемое внутренним профилировщиком, позволяет правильно распределять ресурс процессора между отдельными моделями, делать заключения о соответствии той или иной модели требованиям РВ.

Для контроля и управления параметрами модели используются служебные табличные форматы, позволяющие в режиме РВ вызывать на индикацию или изменять любые переменные модели. Предусмотрен вывод на контроль переменных любых типов, а также их массивов (полностью или в любом заданном диапазоне индексов). Имеется возможность задавать изменение любой переменной скачком или с заданной скоростью нарастания, что позволяет имитировать различные тестовые сигналы.

Кроме того, любая переменная или список переменных модели может быть выведен на отдельный график с последующим документированием.

Подводя итог обзора имеющихся в ЭНИКАД средств разработки и отладки модели РВ, можно констатировать, что создана инструментальная система, позволяющая создавать адекватные модели практически любых технологических объектов, включая такие сложные, как атомные энергоблоки. Например, разработанный на базе ЭНИКАД анализатор для 2-го энергоблока Калининской АЭС с ВВЭР-1000 обеспечивает точность моделирования ТП сравнимую с точностью измерений на реальном энергоблоке. Модель воспроизводит поведение объекта в широком спектре штатных режимов, а также при различных отказах оборудования и сложных аварийных режимах типа разрыва главных паропроводов АЭС.

Другой частью проектной работы является создание набора интерфейсных форматов. Разумеется, дизайн, число и функциональность таких форматов полностью диктуется требованиями проекта. Однако общие принципы организации интерфейса остаются постоянными — любое взаимодействие с моделью осуществляется с помощью тех или иных динамических элементов, размещаемых на формате. В распоряжении разработчика имеется широкий набор таких элементов.

Для организации разнообразных управляющих воздействий с помощью мыши используется универсальный механизм в виде зон управления. Зона управления может иметь прямоугольную или эллиптическую форму. Зоны управления могут составлять двухмерную прямоугольную, треугольную или даже радиальную решетку. Зоны могут быть независимы или зависимы друг от друга. В любом случае, щелчок мыши внутри зоны вызовет изменение переменной модели, тем самым, сигнализируя о наступлении некоторого события. Обычно зона управления не имеет визуального представления, предоставляя разработчику право самостоятельно изобразить поверх зоны адекватный орган управления. Частным случаем зон управления являются кнопки разных видов. Имеются специализированные зоны управления для выполнения служебных функций: перехода на другой формат, вызова подсказки и т. д.

При необходимости передать программе числовой или символьный параметр, разработчик может разместить на формате строку ввода. Размер, ориентация, шрифт, цвет – все поддается настройке.

Богатый выбор предоставляется разработчику для отображения состояния модели. Для экранного интерфейса применимы разные стили оформления: имитация фрагментов пульгов и панелей, изображения технологических схем с управляемыми мнемознаками, копии экранов управляющих компьютеров. Примеры форматов и их элементов приведены на второй полосе обложки. Базовая библиотека элементов включает компоненты любых известных приборов: стрелочных, показывающих, цифровых. Существующие библиотеки мнемознаков охватывают мнемоники, используемые в практике атомных и тепловых электростанций разных типов. Сложные динамические графические объекты создаются в виде групп графических примитивов. Их можно включать в любые графические форматы ЭНИКАД.

Графический редактор позволяет объединять в группы элементы разных типов. Это дает возможность создания типовых интерфейсных конструкций. Интересно, что группы элементов можно сделать управляемыми от модели, изменяя их взаимное расположение и видимость. Так организуются простейшие формы анимации.

Любой динамический элемент становится связанным с моделью после назначения ему специального свойства – интерфейсного имени, в форме технологического имени или программного идентификатора. Интересной особенностью графического модуля GIW является способность совмещения редактирования и отображения состояния элемента от модели. Эта функция особенно полезна при окончательной доводке интерфейса.

Но вот, разработка проекта завершена. Теперь на сцену выходят другие программы ЭНИКАД, призванные облегчить использование прикладной системы. Исполняющей системой по-прежнему является GIW с заблокированными функциями редактирования, но его запуск производится из специальной оболочки – планировщика заданий.

Для чего нужен планировщик заданий? Представим себе, что мы сделали обучающую систему – тренажер. Процесс обучения организуется в форме курса из ряда занятий. Каждое занятие состоит из сеанса работы на тренажере, настроенного на решение вполне определенной учебной цели. Настройка включает определение круга имитируемого оборудования, начальных условий, цели, условий достижения цели, параметров для контроля качества управления ТП и т. д. Вся совокупность описаний такого

рода, представленная в виде конфигурационных параметров GIW, оформляется в форме задания.

Планировщик заданий позволяет накапливать разработанные и проверенные задания в общей базе. В базе планировщика сохраняются сопутствующие файлы: сценарии, списки протоколируемых переменных, а также результаты тренировок в форме протоколов.

Функцией планировщика заданий является также организация обучения в компьютерных классах. Инструктор имеет возможность централизованно запускать произвольные задания на любых компьютерах класса. При необходимости компьютер инструктора может быть подключен в режиме монитора к тренажеру, выполняющему задание. Это позволяет производить скрытое наблюдение за операциями обучаемого, вносить необходимые коррективы или, наоборот, вводить неожиданные возмущения или отказы.

После завершения задания наступает этап оценивания результатов тренировки. В качестве источника информации используется протокол, накопленный в результате занятия. Просмотр протокола инициируется из архива протоколов планировщика заданий. Анализ протокола производится в специальной программе просмотра протоколов – последней из рассматриваемых частей ЭНИКАД.

Протокол предоставляет информацию об изменении значений переменных во времени, событиях, связанных с изменениями значений переменных, и системных событиях.

Протокол в общем случае состоит из нескольких фрагментов, называемых траекториями. Начало каждой траектории всегда связано с запуском модели с новыми начальными условиями. Завершение траектории может быть вызвано операцией отката, загрузкой новых начальных условий или завершением сеанса работы.

Программа просмотра протоколов обеспечивает навигацию между траекториями, позволяет просматривать списки событий и изменения переменных, представленных в виде графиков. При необходимости, различные компоненты протокола могут быть распечатаны на принтере или экспортированы в другие программы для составления отчета.

С момента, когда были написаны первые строки программного кода ЭНИКАД, прошло более 10 лет. За это время комплекс прошел путь взросления, шлифовки на крупных проектах. В итоге удалось создать лицензионно чистый российский продукт, не уступающий лучшим зарубежными образцам. Разработчики и по сей день продолжают совершенствовать программы ЭНИКАД, отражая в них новейшие тенденции и технологии, появляющиеся в мире Windows.

*Чернаков Виктор Алексеевич – директор,
Выговский Сергей Борисович – канд. техн. наук, зам. директора,
Осадчий Михаил Андреевич – ст. научн. сотрудник,
Кориковский Константин Петрович – ст. научн. сотрудник.,
Краюшкин Юрий Викторович – ст. науч. сотрудник компании ЭНИКО ТСО.
Контактный телефон (095) 323-95-99, факс 324-09-93. E-mail:contact@eniko.ru <http://www.eniko.ru>*