

КОНЦЕПЦИЯ РАЗРАБОТКИ ОТКРЫТЫХ ИНТЕЛЛЕКТУАЛЬНЫХ ПРОГРАММНЫХ СИСТЕМ СРЕДНЕГО КЛАССА ДЛЯ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ ЯДЕРНЫХ РЕАКТОРОВ

Зизин М.Н.

РНЦ-КИ, ИЯР

Аннотация

Предложена концепция разработки системного и прикладного программного обеспечения реакторных расчетов. Концепция предусматривает автоматизированную разработку программ, обеспечивающих расчет реакторных характеристик при наличии удобной для пользователя среды. Наиболее полно концепция применялась при создании Windows-версии интеллектуальной программной оболочки ShIPRW (Shell of Integrated Package for Reactors for Windows) для математического моделирования ядерных реакторов. Разработанное системное наполнение может использоваться и в других предметных областях.

Ключевые слова: Математическое моделирование, база знаний, нейтроника, модульные системы

The summary

Concept of developing the system and application software for reactor computations is offered. It provides automated development of the codes ensuring a user-friendly computation of the reactor characteristics. The concept was most completely used while creation of the ShIPRW intellectual code shell (Shell of Integrated Package for Reactors for Windows) for mathematical simulation of nuclear reactors. The developed system software can be used in other application fields.

Keywords: Mathematical Simulation, Knowledge Base, Neutronics, Modular Systems

1 Введение

Бурно развивающаяся область компьютерной программной технологии (software engineering) - инженерия знаний (knowledge engineering) - позволяет заново взглянуть на традиционные задачи математического моделирования таких сложных установок как ядерные реакторы.

Сегодня самой передовой технологией для организации работы с большими наборами программ являются системы, в той или иной мере использующие формализованные знания. В задачах математического моделирования их

можно использовать для построения расчетных комплексов программ на основе библиотек вычислительных модулей, для их эффективной эксплуатации, а также для облегчения работы с ранее созданными большими и сложными пакетами прикладных программ.

С другой стороны, в настоящий момент нужно исходить из необходимости максимального использования уже проведенных научных разработок в области математического моделирования ядерных реакторов, с дальнейшим постепенным накоплением знаний. Раньше, как правило, разрабатывались достаточно большие программы и программные комплексы, которые обычно не поддавались расчленению. В этом случае при решении вновь возникающих задач приходилось заново программировать многие алгоритмы. Сегодня предлагается сосредоточить усилия на производстве отдельных вычислительных модулей (в значительной мере из уже готовых программ) с одновременной разработкой системного обеспечения и базы знаний, работающих с создаваемыми библиотеками вычислительных модулей.

Предлагаемая концепция создания системного и прикладного программного обеспечения реакторных расчетов выросла из попыток обобщить на современном уровне опыт создания программных комплексов и модульных систем для расчета реакторов. Концепция предусматривает разработку программ, обеспечивающих автоматизированный расчет физических характеристик реальных компоновок реакторов, при наличии удобной для пользователя среды, что является актуальной задачей, нужной при всех типах расчетов

Наиболее полно разработанная концепция использовалась при создании открытой интеллектуальной программной системы среднего класса ShIPRW – WINDOWS-версия оболочки ShIPR^{1,2,3} (Shell of Integrated Package for Reactors) для математического моделирования ядерных реакторов. К среднему классу относятся системы, создаваемые небольшим коллективом разработчиков – от 2 до 5 человек. В этом случае облегчается управление проектом и исключаются многочисленные формальности, требуемые при разработке больших проектов. Реализация систем среднего класса существенно упрощается, если в коллектив входят и пользователи системы.

2 Концепция

Постоянно совершенствуемая концепция позволяет оценить любой сделанный или находящийся в разработке программный продукт и наметить перспективу его улучшения.

Сформулируем основные положения концепции создания системного и прикладного программного обеспечения для задач математического моделирования (курсивом выделены пока не реализованные в системе ShIPRW пункты). По сравнению с публикацией⁴ концепция структурирована и существенно доработана^{5,6}.

2.1 Системное наполнение

Цели:

- Автоматизированная генерация больших программ (стандартных путей расчета) на языке Фортран с использованием библиотек вычислительных модулей и соответствующей базы знаний о предметной области и программной среде (технология сборочного программирования).
- Планирование распараллеленного на уровне модулей вычислительного процесса с возможностью его выполнения в многопроцессорной и/или распределенной среде.
- Обеспечение накопления знаний о предметной области в виде описаний программ, их интерфейсов, типовых наборов исходных данных и правил использования и связывания объектов базы знаний. Наследование связей объектов.
- Предоставление пользователю возможности работы на основе примеров.
- Совместное использование нескольких однотипно сконструированных систем для связанных предметных областей.
- Профилирование информации в соответствии с решаемой задачей и/или требованиями пользователя.
- Возможность работы с графическими объектами.
- Интеллектуальная контекстно-зависимая помощь для ускорения работы и уменьшения числа ошибок пользователя.
- Реализация сценариев работы с разной степенью подробности в зависимости от уровня подготовки пользователя и/или степени отлаженности решаемой задачи.
- Работа на вычислительных комплексах разной мощности, начиная с младших моделей рабочих станций.

Подсистемы:

- Использование различных языков программирования при реализации подсистем.
- Фронтальная подсистема, использующая для подготовки входной расчетной информации наборы типовых данных и базу знаний по вычислительным модулям и их интерфейсам.
- Подсистема, обеспечивающая работу с правилами использования модулей, их интерфейсов и других системных объектов.
- Справочная подсистема, работающая в онлайн-овом и контекстно-зависимом режимах.
- Система обработки ошибок пользователя, предлагающая ему варианты их устранения.
- *Основанные на знаниях подсистемы для оказания интеллектуальной помощи пользователям в ранжировании задачи от простого к сложному - интеллектуальный поводыр.*

2.2 Функциональное наполнение

- **Накопление процедурных знаний в виде библиотеки вычислительных модулей вместе со знаниями об их использовании с применением правил. Накопление типовых наборов исходных данных для модулей и их сочетаний. Модуль - это подпрограмма на языке Фортран-77.**
- **Создание среды, облегчающей отладку и модификацию вычислительных модулей, их использование и сопровождение. Поддержка процесса верификации программ и их совокупностей в течение их жизненного цикла.**
- **Использование inline модулей, автоматически встраиваемых в текст головной программы.**
- **Создание предметно-ориентированных подсистем - например, визуализации входной и выходной информации.**
- **Перенесение в базу данных ряда действий, иногда выполняемых вычислительными модулями (подготовка информации, работа с объектами, списками и структурами).**
- **Отчуждение знаний и как частный случай - отчуждение программ от разработчика.**
- **Создание сценариев событий и/или последовательностей расчетов. Генерация входных данных на основе сценариев.**

2.3 Документация

- **Документирование системы на всех этапах жизненного цикла.**
- **Наличие качественной документации для системных и прикладных подсистем.**

2.4 Требования

- **Максимальная открытость и гибкость программного обеспечения с легким включением как новых вычислительных модулей, так и новых системных программ и подсистем.**
- **Открытая компонентная архитектура. Модульность при разработке прикладного и системного программного обеспечения, допускающая наращивание и модификацию.**
- **Возможность работы как в сети, так и на отдельном компьютере.**
- **Дружественный интерфейс с пользователем на всех этапах создания и эксплуатации программных продуктов. Облегчение работы среднего пользователя, не обладающего глубокими знаниями функционального и системного наполнения.**
- **Адаптируемость к изменяющимся требованиям пользователей.**
- **Устойчивость работы программного обеспечения при неверных действиях пользователя.**
- **Быстрота освоения системы пользователем. Простые правила применения функциональности для конечного пользователя.**
- **Создание базы знаний о предметной области, программной среде и тестовых задачах.**

- Широкое применение правил использования системных и функциональных объектов.
- Хранение в базе знаний наборов повторно используемых данных – как в формате NAMELIST'а, так и в виде любых готовых исходных данных для отдельных программ.
- Использование баз данных для генерации входной информации.
- Возможность применения объектно-ориентированных технологий.
- *Ранжирование информации для пользователя. Обеспечение средств обязательного донесения до пользователя важной информации.*
- *Автоматическое накопление количественных, частотных и временных данных о работе прикладного и системного программного обеспечения.*
- *Лексический анализ словесных определений объектов для:*
 - *определения принадлежности объекта к соответствующему разделу предметной области;*
 - *образования новых стандартных величин по определенным правилам;*
 - *выделения ключевых слов и поиск объектов по ним.*
- Возможность совершенствования в процессе эксплуатации с изменением функций при изменении требований со стороны пользователей.
- Возможность итерационного развития с последовательным созданием новых версий программной системы.
- Эволюционное наследование баз знаний, свойств и компонентов систем при переходе на новые средства реализации.
- Участие заказчика (потребителя) в формировании требований к программному продукту с начала разработки систем.
- Отчуждаемость системы от разработчиков с возможностью освоения квалифицированными пользователями.

2.5 Вычислительная среда

- Максимальная машинная независимость создаваемых программ.
- Возможность работы как на любой одиночной ЭВМ, так и в разнородной компьютерной сети, содержащей персональные компьютеры, рабочие станции, мейнфреймы и супер-компьютеры.
- Использование персонального компьютера или рабочей станции для всех этапов работы, кроме очень больших вычислений.

3 Интеллектуальная программная система ShIPRW для математического моделирования ядерных реакторов

Состояние дел в области математического моделирования близко к тому, чтобы данные и программы, хранимые в ЭВМ, использовались как знания при решении расчетных задач. В базу знаний по математическому моделированию могут входить следующие составные части:

1. Библиотека вычислительных (процедурные знания).
2. Список стандартных величин, в терминах которых описываются интерфейсы вычислительных модулей (межмодульный интерфейс - входные и выходные данные вычислительных модулей).

3. **Информация, необходимая для организации вычислительного процесса. Знания о вычислительной среде - операционных системах и ЭВМ, с помощью которых должны выполняться программы.**
4. **Типовые наборы исходных данных как для отдельных вычислительных модулей, так и для их совокупностей.**
5. **Знания о правилах использования вычислительных модулей, методах расчета и ограничениях на их применимость. Эти знания могут включать в себя как количественную, так и качественную информацию.**
6. **Библиотеки физических констант и свойств материалов.**

Два компонента базы знаний - библиотека и межмодульный интерфейс (набор стандартных величин - СВ) - могут быть без изменений использованы практически в любых вариантах программного обеспечения для математического моделирования сложных объектов. При решении вновь возникающих задач можно будет максимально использовать уже имеющиеся вычислительные модули.

В рамках реализации предложенной концепции создана интеллектуальная программная система для задач математического моделирования ядерных реакторов ShIPRW, обеспечивающая автоматизированную генерацию головных программ на языке ФОРТРАН-77 на основе библиотеки вычислительных модулей и соответствующей базы знаний.

Создаваемые стандартные пути (СП) расчета (головные программы), содержащие вызовы цепочек вычислительных модулей, могут выполняться практически на любой ЭВМ.

В создании системной части ShIPR на разных этапах ее разработки участвовали Сушнова Н.Б., Шушаков А.В., Дементьева Д.Ю., Богданкевич В.Л. и Тихонова Н.В. WINDOWS-версия оболочки ShIPR – ShIPRW – создавалась автором совместно с Сушновой Н.Б., причем большая часть реализации оболочки средствами Visual FoxPro сделана Сушновой Н.Б.

По отношению к ShIPRW употребляется два определения - система и оболочка. Термин система используется при ссылке на все составные части ShIPRW, а термин оболочка может не включать в себя ядро NUCLEUS и/или процедурные знания в виде библиотеки вычислительных модулей.

Система ShIPR первоначально создавалась и долгое время эксплуатировалась в среде MS DOS. В 1998-99 гг. была создана WINDOWS-версия системы ShIPRW. При ее создании особое внимание уделялось необходимости сохранения базы знаний из DOS-версии ShIPR. Все нижеописанное относится уже к WINDOWS-версии системы, если не оговорена ссылка на DOS-версию.

Основная часть ShIPRW создана в среде FoxPro. Часть планировщика PLEX написана на PROLOG'e. Подсистема NUCLEUS реализована на ФОРТРАНе-77.

Система ShIPRW состоит от нескольких довольно независимых частей. Некоторые из этих подсистем могут функционировать отдельно, и некоторые могут использоваться в другой среде. Автономия отдельных частей ShIPRW и возможности их связи в отдельные технологические цепочки облегчают стадии создания, проверки и обслуживания программ.

Ниже описаны особенности отдельных подсистем.

3.1 Система накопления знаний

В базе знаний пока реализовано накопление знаний о следующих основных объе

- предметная область и ее подобласти;
- стандартные величины (сумма входных и выходных данных вычислительных модулей);
- служебные стандартные величины, необходимые для организации вычислительного процесса;
- вычислительные модули;
- наборы данных;
- правила;
- стандартные пути расчета;
- вычислительная среда;
- файлы.

3.2 PLEX - планировщик вычислительного процесса

Планировщик вычислительного процесса PLEX ^{7,8} по поставленной конечной цели на основе заранее представленной и/или полученной в режиме диалога информации строит цепочку вычислительных модулей, приводящую к этой цели. Входной информацией для PLEX является список стандартных величин, которые должны быть рассчитаны и список доступной входной информации. Пользователь имеет возможность выбрать оба списка из полного списка для данной предметной области. Выбор необходимых исходных данных облегчает подготовленный системой список уникальных входов - величин, не рассчитываемых ни одним модулем. При планировании в цепочку могут быть включены рекомендованные модули.

Естественно, что ощутимые выгоды от применения планировщика начинают появляться только после того, как число доступных вычислительных модулей превысит некий порог в несколько десятков модулей. Наибольший эффект можно ожидать при планировании распараллеленного вычислительного процесса в многопроцессорной вычислительной среде. В связи с распространением вычислений в распределенных средах в качестве разных процессоров могут выступать другие компьютеры в сети.

Наличие достаточно большого числа вычислительных модулей тесно связано с принципами модуляризации. Гигантомания в размерах и функциях вычислительных модулей делает практически ненужным планирование вычислительного процесса, но она же и лишает возможности эффективно

распараллелить вычислительный процесс и затрудняет модифицирование алгоритмов. К тому же в этом случае от пользователя требуется хорошее знание возможностей каждого крупного вычислительного модуля, что затрудняет эксплуатацию систем с крупными модулями.

3.3 NUCLEUS - ядро системы SHIPRW

Функция NUCLEUS³ - автоматизированное получение текста машинно-независимой головной программы на языке Фортран-77, содержащей вызовы заданного или построенного планировщиком набора вычислительных модулей. Его работа строится на основе специально подготовленной информации об интерфейсах вычислительных модулей и некоторой служебной информации.

В состав ядра входят программы, предназначенные для автоматизации построения стандартных путей расчета на основе списков стандартных величин – интерфейсов для модулей - и самих вычислительных модулей, участвующих в расчете. Получаемый текст стандартного пути является машинно-независимым и практически готов к исполнению на любой машине.

3.4 Стандартные пути расчета

Основной программной единицей, с которой имеет дело пользователь, производящий конкретные расчеты, является стандартный путь расчета. СП - это головная программа (MAIN в Фортране), которая содержит обращения к вычислительным модулям.

Кроме вызова вычислительного модуля, стандартный путь содержит описания всех стандартных величин, операторы EQUIVALENCE и обращения к программам работы с файловым архивом и программам печати. При появлении новых задач, не покрываемых доступными СП, пользователь может найти в имеющихся СП достаточно близкий к его целям прототип и модифицировать его для своей задачи. Каждому стандартному пути соответствует свое подмножество списка стандартных величин.

Реализована следующая схема конструирования СП:

1. Создается набор таблиц, содержащий списки всех стандартных величин, используемых в СП, и их основные характеристики. Кроме набора таблиц, в исходную информацию входит и список вызываемых вычислительных модулей. По этому списку выбирается стандартизированное обращение к вычислительному модулю, содержащее в качестве фактических параметров только стандартные величины, описанные в таблицах.
2. Наличие указанных таблиц является достаточным для автоматизированного построения текста стандартного пути. Исходной информацией для этого служат заготовки некоторых фрагментов текста и таблицы со служебной информацией для данного СП.

3.5 Межмодульный интерфейс - список стандартных величин

Для удобства формального описания функционального и системного наполнения вводится понятие стандартной величины (СВ). В терминах функциональных СВ можно определить вход и выход любого вычислительного модуля, а в терминах служебных СВ в основном описываются атрибуты функциональных СВ, необходимые для организации вычислительного процесса.

Список стандартных величин является открытым и может постоянно пополняться. Для каждой стандартной величины вводится стандартное обозначение, под которым понимается идентификатор СВ вместе со списком ее индексов и их границ.

Для стандартных величин, отличающихся только длиной слова (одинарная или двойная точность), рекомендуется использовать разные идентификаторы.

Для каждой стандартной величины хранятся следующие атрибуты - идентификатор СВ, тип элемента СВ (INTEGER, REAL, DOUBLE PRECISION, CHARACTER, LOGICAL), длина элемента в байтах (для CHARACTER), индексы СВ-массивов, формат элемента СВ для операций ввода/вывода, предметная область, единица измерения, словесное описание элемента СВ, границы для значений СВ. При описании стандартного пути дополнительно указываются имена файлов для хранения СВ и ключи использования СВ-массивов (печать, чтение из архива, запись в архив, тип массива).

Наборы стандартных величин предназначены для решения задач определенной предметной области.

Создание и функционирование отдельных стандартных путей расчета сопровождается созданием набора служебных стандартных величин, которые содержат атрибуты функциональных СВ.

Библиотека вычислительных модулей характеризуется набором стандартных величин, участвующих в расчете. В терминах стандартных величин можно определить вход и выход любого вычислительного модуля. Список СВ является открытым и может постоянно дополняться. Допустимо наличие нескольких стандартных величин для одних и тех же физических и математических величин, отличающихся внутренней структурой массива и/или размерностью.

Использование стандартных обозначений для стандартных величин и закрепленных обозначений для индексов в тексте модуля является желательным, но не обязательным. Использование СВ в качестве фактических параметров в среде ShIPRW является обязательным.

Предполагается, что для вычисления конкретной стандартной величины может иметься несколько различных вычислительных модулей.

Списки стандартных величин создаются для решения задач определенной предметной области. Для разных предметных областей списки СВ могут частично перекрываться.

3.6 Вычислительные модули

Набор вычислительных модулей, входящих в конкретную библиотеку, должен в пределе покрывать расчет всех стандартных величин, не являющихся исходными данными. Каждый вычислительный модуль рассчитывает одну или несколько стандартных величин. Для расчета одной СВ может существовать несколько вычислительных модулей.

Кратко требования к вычислительному модулю можно изложить следующим образом:

1. Вычислительный модуль - подпрограмма, написанная на языке Фортран-77.
 2. Все входные данные и результаты расчета передаются через параметры подпрограммы, без использования COMMON-блоков. Использование COMMON-блоков не рекомендуется по следующим причинам:
 - массивы в COMMON-блоках требуют статического распределения памяти;
 - структура COMMON-блока определяется в модулях, их использующих, что при отсутствии согласованности испытаний является постоянным источником ошибок;
 - внутри одного модуля используется, как правило, не весь COMMON - блок, что затрудняет понимание модуля;
 - использование для организации межмодульных интерфейсов параметров, а не COMMON-блоков, позволяет отделить входные данные и результаты отдельного вычислительного модуля от интерфейса всего комплекса программ, в который может войти вычислительный модуль. Это делает модуль законченным продуктом, отчужденным от конкретного окружения.
 3. Не рекомендуется использовать инструкцию EQUIVALENCE, потому что она препятствует оптимизации программ.
 4. Вычислительный модуль обладает только одним входом.
 5. Рекомендуемый размер модуля 60-70 строк текста, но не более 200 строк (не считая строк комментария).
 6. Все вычисления внутри вычислительного модуля и модуля проводятся, как правило, только с использованием оперативной памяти.
 7. Если предполагается, что оперативной памяти может не хватить для расположения всех нужных массивов, дополнительно пишутся альтернативные варианты вычислительного модуля или отдельных подпрограмм, в которых используется подкачка данных из внешней памяти. Рекомендуется это делать с помощью подпрограмм, реализующих прямой доступ к массивам во внешней памяти по именам.
- Вычислительные модули, отвечающие этим требованиям, с

минимальными затратами могут быть включены как в библиотеку вычислительных модулей, так и в другие пакеты программ. Такие требования позволяют вести разработку вычислительных модулей параллельно с разработкой системного наполнения пакетов прикладных программ.

На практике требования к программам, превращаемым в вычислительные модули, резко снижены и модули могут превращаться в подпрограммы достаточно формально, получая всю необходимую им информацию через собственные файлы, подобно тому как это делает программа WIMS.

3.7 Программы печати стандартных величин

Программы печати массивов позволяют распечатывать целые, вещественные и символьные массивы размерностью не более 4. На каждой странице в заголовке указывается номер страницы, имя печатаемого массива и его размер. Каждый столбец и строка имеют в заголовке соответствующий номер.

Программа печати скаляров позволяет печатать СВ-скаляры вместе с их словесным описанием.

3.8 Фронтальная подсистема FSW

Фронтальная подсистема FSW обеспечивает подготовку входных данных для конкретного стандартного пути расчета, их проверку и самосогласованность. Подсистема FSW может быть использована и как фронтальная система для ранее созданных комплексов программ при условии заполнения соответствующей базы знаний.

FSW запрашивает у пользователя недостающую информацию, анализирует ее на предмет применимости данного пути для решения конкретной задачи и готовит набор исходных данных, с помощью которых может уже быть произведен окончательный расчет. Такой подход позволяет уменьшить непроизводительные траты машинного времени, расходуемого на расчеты либо плохо подготовленных вариантов, либо на расчеты по программам, не соответствующим поставленным задачам.

3.9 Наборы данных

Назначение наборов данных - объединение достаточно разнородной информации, описывающей какой-либо объект. Обычно набор данных является частью математической модели. Пример - геометрические характеристики реактора. Конечная цель набора данных - задание перед расчетом набора числовых значений, хранящихся в базе данных. В последнем случае, передав один набор данных, мы получаем значения целого набора стандартных величин - для одного или нескольких модулей.

Использование наборов данных существенно облегчает подготовку данных для расчета за счет постепенного уменьшения числа ошибок в данных и

возможности использования наборов данных в разных стандартных путях расчета. В стандартном пути обычно используется несколько наборов данных.

3.10 Верификация и применение

Система ShIPRW и ее библиотека вычислительных модулей тестировалась с помощью большого набора тестов ANL-7416⁹, а также тестов для реакторов БН-600 и БН-800. Результаты расчета тестов ANL-7416 переданы в ОФАП ФЭИ.

Неофициальную апробацию система ShIPR прошла, когда автор 'вслепую' получил хорошо совпадающие результаты для европейского теста EFR, созданного для изучения проблемы накопления транснуклидов в быстрых реакторах.

В течение последних двух лет с помощью системы ShIPR автором проводились многосерийные нейтронно-физические расчеты по российско-американскому проекту 6320 высокотемпературного графитового реактора GT-MHR. Расчеты проводились по константам, подготавливаемым системами MCU и WIMS.

Система ShIPR находится в эксплуатации в Физико-энергетическом институте, г. Обнинск, и Московском Высшем Техническом Университете.

Есть опыт передачи отдельных стандартных путей, сгенерированных с помощью системы. Переданные стандартные пути могут эксплуатироваться как отдельные программы.

В настоящее время система ShIPRW используется для расчетов тестов математической модели 4-го блока Чернобыльской АЭС¹⁰. Данная работа проводится по контракту МНТЦ 0683, целью которого является создание набора тестовых задач, моделирующих предаварийное состояние 4-го блока ЧАЭС.

4 Заключение

На основе разработанной концепции создана интеллектуальная программная оболочка (система) ShIPR для математического моделирования ядерных реакторов.

Для создания отдельных частей системы использовались языки FoxPro, Фортран-77, Prolog. Оболочка работает на PC/486 и выше. Система эксплуатируется и совершенствуется в течение нескольких лет и в настоящее время работает версия ShIPRW 6.01.

Существующая библиотека вычислительных модулей ShIPRW включает в себя модули, реализующие основные этапы нейтронно-физического расчета реакторов - расчет макро- и микросечений, трехмерных нейтронных полей, расчет реактивностей и изменения изотопного состава топлива.

Усовершенствование интеллектуальной программной оболочки ShIPRW возможно по следующим направлениям:

- Увеличение числа атрибутов для некоторых объектов.
- Дополнение базы знаний новыми объектами.
- Создание новых подсистем - накопление знаний на основе вычислительных экспериментов, векторизация.
- Подключение специализированных проблемно-ориентированных подсистем - подготовки информации для описания геометрии расчетной модели реактора и составов зон и визуализации результатов расчета.
- Оптимизация использования оперативной памяти.

Система может применяться теми, кто хочет:

- привести в порядок большое программное хозяйство на языке Фортран;
- повысить производительность труда прикладных программистов;
- стать независимым от авторов программ и конкретных типов компьютера;
- эволюционным способом перейти к современным методам работы - от устаревающего комплекса программ к интеллектуальной модульной системе или просто с большими удобствами работать с трудным в эксплуатации комплексом программ.

Разработанное системное наполнение может использоваться и в других предметных областях.

Моделироваться могут любые объекты и процессы, описываемые набором вычислительных модулей (процедурные знания). Перед моделированием заполняется база знаний о модулях и их интерфейсах.

4.1 Основные достоинства системы ShIPRW

- Формирование текста головной программы на языке Фортран-77 и подготовка к расчету на персональном компьютере с возможностью выполнения программы на любых ЭВМ, имеющихся в сети.
- Удобство среды для разработки, отладки и хранения вычислительных модулей и их совокупностей.
- Помощь пользователю при задании исходных данных для расчета, с проверкой их на разумность.
- Дружественная среда для накопления знаний о вычислительных модулях и их интерфейсах с отчуждением программ от их авторов.
- Встроенная системная и предметно-ориентированная документация.
- Открытость функционального наполнения.
- Относительная независимость отдельных частей системного наполнения.
- Возможность использования программной оболочки ShIPRW для различных предметных областей.

4.2 Что дает оболочка ShIPRW пользователю

- Возможность автоматической генерации больших программ (стандартных

путей расчета) на языке Фортран из заданной последовательности вычислительных модулей.

- Работа в единой среде.
- Наглядность, гибкость, относительная простота, работа на основе примеров.
- Применение персонального компьютера для всех этапов работы, кроме очень большого счета.
- Возможность постепенного освоения системы – от простых расчетов по готовым стандартным путям расчета до выхода на уровень суперпользователя, способного использовать систему ShIPRW для решения нестандартных задач. Для молодых и рвущихся система предоставляет большие возможности продемонстрировать свое умение.
- Применение в стандартных путях расчета наборов повторно используемых данных в формате NAMELISTа, что существенно облегчает подготовку данных для расчета. Наборы данных овеществляют знания авторов программ и опытных пользователей.
- Резкое уменьшение числа ошибок при расчете на основе:
 1. обнаружения ошибок на начальных этапах подготовки данных с помощью правил использования вычислительных модулей и их интерфейсов;
 2. раннего выявления неправильной последовательности модулей;
 3. диагностирования отсутствия необходимой входной информации;
 4. возможности задания данных группами с выверенными исходными данными;
 5. уменьшения объема данных, которые должен задавать пользователь за счет использования наборов данных, правил и задания значений ряда констант и служебных параметров по умолчанию.

Широкое применение фильтров по предметным областям и группам стандартных величин существенно облегчает выбор вычислительных модулей, стандартных величин, стандартных путей расчета, файлов и наборов данных.

Накопление знаний в виде отлаженных стандартных путей расчета и выверенных наборов входных данных для неоднократно используемых сочетаний данных.

Использование баз данных для генерации входной информации.

Генерация перед расчетом вспомогательного файла со словесными определениями всех входных данных.

Наборы рекомендованных данных для программ со сложными наборами входной информации.

Возможность работы в привычном режиме с использованием NAMELIST'а.

Обеспечение адекватных исходных данных при реализации альтернативных способов расчета.

Возможность накопления большого набора тестовых задач.

Пользователь-непрограммист может сам создавать стандартные пути расчета, содержащие новые цепочки из имеющихся вычислительных модулей.

**Решение нестандартных расчетных задач за счет:
открытости и гибкости системы,
относительной легкости включения новых модулей,
реальной возможности использования результатов расчета по другим комплексам программ.**

Использование постоянно пополняемого набора вычислительных модулей.

Облегчение отладки и модификации вычислительных модулей, их использования и сопровождения. Поддержка процесса верификации программ и их совокупностей в течение их жизненного цикла.

Программист может создавать и отлаживать новые программы, работающие в сочетании с уже имеющимися модулями, которые обеспечивают для вновь создаваемой программы входную информацию и расчет различных функционалов.

Самым простым способом создания небольших новых модулей является использование in-line модулей, автоматически встраиваемых в текст головной программы. Такой подход позволяет создавать программы без их включения в библиотеку объектных модулей и без описания СВ-массивов в тексте программы.

Легкая возможность создавать новые сервисные модули для предварительной обработки входной информации и представления результатов в нужном для анализа и публикаций виде.

Использование текстов имеющихся программ как прототипов при создании новых модулей, что особенно эффективно при создании сервисных модулей.

Автоматизированное создание супер-модулей – объединения часто повторяющихся в стандартных путях расчета совокупностей подряд идущих модулей. Список параметров супер-модуля образуется из суммы параметров составляющих модулей с автоматическим определением входов и выходов.

Уверенность в правильности результатов при использовании альтернативных способов расчета, когда в системе есть различные модули для расчета одинаковых величин.

Литература

- ¹ Зизин М. Н. Системное и прикладное программное обеспечение задач математического моделирования ядерных реакторов. Докторская диссертация, РНЦ-КИ, 1995
- ² Зизин М.Н., Шушаков А.В., Дементьева Д.Ю., Сушнова Н.Б. Интеллектуальная программная оболочка ShIPR для математического моделирования ядерных реакторов. Общее описание. Препринт ИАЭ-5701/5, М., 1994.
- ³ Зизин М. Н., Шушаков А.В., Дементьева Д.Ю. Интеллектуальная программная оболочка ShIPR для математического моделирования ядерных реакторов. NUCLEUS - ЯДРО СИСТЕМЫ ShIPR. Препринт ИАЭ-5867/4, М., 1995.
- ⁴ Зизин М.Н. Концепция создания системного и прикладного программного обеспечения реакторных расчетов. Атомная энергия, т.75 вып 5, май 1995 с.299-305
- ⁵ Зизин М.Н. Интеллектуальное программное обеспечение задач математического моделирования. Препринт РНЦ-КИ, ИАЭ, 6108/5, М., 1999
- ⁶ Zizin M.N. The Concept of Creation of the System and Application Software for Reactor Computations and It Implementation in an Intellectual Program Shell ShIPR. Preprint RRC-KI, IAE 6108/5, M., 1999
- ⁷ Сушнова Н.Б., Зизин М.Н. Экспертный планировщик PLEX для планирования вычислительного процесса в среде интеллектуальной программной оболочки математического моделирования ядерных реакторов ShIPR. Препринт РНЦ-КИ, ИАЭ 6044/5, 1997.
- ⁸ Сушнова Н.Б. Алгоритмы планирования вычислительного процесса для экспертного планировщика PLEX. Препринт РНЦ-КИ, ИАЭ 6043/5, 1997
- ⁹ Argonne Code Center: Benchmark Problem Book. Numerical Determination of the Space, Time, Angle or Energy Distribution of Particles in an Assembly. National Energy Software Center, Argonne National Laboratory, ANL-7416, 1968, rev. 1972, 1977, 1985
- ¹⁰ Расчет начальной загрузки 4-го блока Чернобыльской АЭС в среде интеллектуальной программной системы ShIPRW Авт. Зизин М.Н., Иванов Л.Д. Сушнова Н.Б. и др. Отчет РНЦ КИ № 36/15, 1999