

Национальная академия наук Украины
Институт кибернетики имени В. М. Глушкова

На правах рукописи

ВАЛЬКЕВИЧ Татьяна Арнольдовна

УДК 51:681.3.06

**ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ПОДДЕРЖКИ
ОБРАБОТКИ ОНТОЛОГИЧЕСКОЙ ИНФОРМАЦИИ
ПРОГРАММНЫХ СТРУКТУР
МНОГОКОМПОНЕНТНОГО ПРОГРАММИРОВАНИЯ**

05.13.11 — математическое и программное обеспечение
вычислительных машин, комплексов, систем и сетей

Автореферат диссертации на соискание ученой степени
кандидата физико-математических наук

Киев 1994



00756881 (Z)

Работа выполнена в Институте кибернетики имени В. М. Глушкова НАН Украины.

Научные руководители: доктор физико-математических наук, профессор КАПИТОНОВА Ю. В.
кандидат физико-математических наук МИЩЕНКО Н. М.

Официальные оппоненты: член-корреспондент АН Украины ЮЩЕНКО Е. Л.,
кандидат физико-математических наук ГРИНЧЕНКО Т. А.

Ведущая организация: Киевский национальный университет имени Т. Г. Шевченко.

Защита состоится «23 сентября 1994 г. в 11 часов на заседании специализированного ученого совета Д 016.45.01 при Институте кибернетики имени В. М. Глушкова НАН Украины по адресу:

252650 Киев ГСП 22, проспект Академика Глушкова, 40.

С диссертацией можно ознакомиться в научно-техническом архиве института.

Автореферат разослан «11 марта» 1994 г.

ЛНБ ім. В. Стефаніка
АН України

Ученый секретарь
специализированного ученого совета

СИНЯВСКИЙ В. Ф.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. Новый уровень программирования, связанный со всеобщей компьютеризацией и информатизацией, вызывает сложные проблемы организации эффективного решения задач в компьютерной среде.

Характер современной программистской деятельности изменился в связи с тем, что за несколько десятилетий ее развития в мире накоплен огромный потенциал программного обеспечения. Единицей программирования стала программная компонента, и программист прекращает программировать изначально и начинает собирать (монтировать, компоновать, склеивать) решение своей задачи из каталогов повторно используемых программных компонент. Интеллектуальный фокус программирования сосредоточен теперь на понимании, выборе и правильном и эффективном использовании компонент.

Для поддержки покомпонентной практики построения программных систем необходимы инструментальные средства и технологии, предоставляющие возможности для решения каждой из ее различных задач. Такими инструментальными средствами являются средства инкапсуляции, спецификации, тестирования, повторного использования, максимально возможно более позднего связывания компонент, а также библиотеки замещаемых программных компонент, их документация и визуализация. Технологии покомпонентного программирования поддерживают гибкую композицию повторно используемых программных компонент, создание открытых развиваемых систем, обеспечивающих восприятие их качеств в понятной для пользователя форме.

Правильное и эффективное использование программных компонент требует рассмотрения различных аспектов организации вычислений. Для поддержки технологии композиции произвольных вычислительных активностей развивается понятие координации вычислений, ортогональное вычислению. Одним из важных аспектов координации вычислений является параллелизм. Координационный язык, представляющий координационную модель вычислений, содержит операторы для создания числительных действий и для поддержки связей между ними. Композиция программных компонент требует не только координации вычислений, но и координации информации, которая может отражать различные парадигмы и быть представлена в разных языках и системах программирования. Информация должна быть собрана, трансформирована и передана различным компонентам процесса программирования. Эффективные системы покомпонентного программирования должны

иметь гибкие средства обработки онтологических элементов, лучше всего соответствующих проблеме, рассматриваемой в каждый конкретный момент.

Таким образом, процесс программирования на основе повторно используемых программных компонент приобретает новое содержание; он заключается в разработке сложного интеллектуального продукта и включает не только анализ данных и их трансформацию, но и организацию связи и координацию. В связи с увеличением сложности и масштаба рассматриваемых задач, эволюционным характером построения их программного решения, необходимостью участия человека в процессе его программирования разработка программных продуктов нуждается в моделировании процесса программирования и разностороннем мониторинге среды функционирования программных компонент в процессе всего жизненного цикла программы. Многие формы информации, лежащие в основе анализа процесса, должны быть представлены и интегрированы в среде. Следовательно, чтобы понять программную компоненту, определить и обеспечить условия ее существования и функционирования в сложноорганизованной программной среде, она должна иметь адекватную информационную структуру в этой среде.

Перспективными направлениями разработки программного обеспечения для поддержки покомпонентного программирования являются объектно-ориентированное программирование и мегапрограммирование.

В рамках объектно-ориентированного программирования (ООП) адаптирован и развит сборочный способ программирования, основанный на компонентах — объектах. Объектно-ориентированная технология предлагает эволюционную разработку прикладных программных систем на основе пошагового программирования из предварительно созданного инкапсулированного программного обеспечения и дальнейшего его наращивания, настройки и сборки.

Парадигма мегапрограммирования сформулирована как направление программной инженерии по проблеме увеличения масштаба обработки очень больших программных систем, охватывающих независимые, разнородные географически распределенные программные компоненты.

Комплексный теоретико-практический подход к проблемам автоматизированной обработки и хранения информации был предложен и развит академиком В.М. Глушковым и его учениками в математической теории проектирования вычислительных систем и подтвержден созданными под их руководством вычислительными системами, такими, как автоматизированная система совместного проектирования схемного и

программного обеспечения ЭВМ и многопроцессорный вычислительный комплекс с макроконвейерной организацией вычислений. В рамках последнего разработан, реализован и оттестирован на ряде сложных задач язык МАЯК, представляющий собой согласованное семейство языков параллельного программирования высшего уровня, ориентированных на поддержку крупноблочных асинхронных координационных схем вычислений в многопроцессорных вычислительных комплексах.

Целью настоящей работы являются построение моделей программных компонент сложноорганизованных программ в распределенной среде вычислений, создание экспериментальных комплексов инструментальных средств для поддержки эффективного покомпонентного программирования распределенных вычислений и их использование в практике построения интеллектуальных трансляторов.

Актуальность направлений исследований работы определяется увеличением сложности процесса программирования, требующего разработки технологий и интеллектуальных инструментальных средств для поддержки эффективного покомпонентного программирования.

В работе решены следующие основные задачи:

- построение моделей параллельно функционирующей программной компоненты крупноблочного программирования в распределенной среде вычислений и разработка языковых средств представления моделей;
- разработка и реализация комплекса инструментальных средств для поддержки информационной обработки программных компонент;
- разработка элементов информационной технологии построения программного решения задач в развитых программных средах.

Методологическую основу исследований составляют результаты математической теории проектирования вычислительных систем, теории структур данных, теории алгебраического программирования, теории параллельного программирования, теории построения трансляторов, объектно ориентированное программирование, а также постановки экспериментов на ЭВМ по созданию инструментальных средств для информационной поддержки программных компонент и их применению.

Научная новизна состоит в разработке методологии использования информации программных компонент в процессе координации вычислений в распределенной программной среде путем построения новых видов модельных ситуаций в процессе покомпонентного программирования распределенных вычислений, а также создании средств программной поддержки этой методологии, эффективность которой подтверждена при создании интеллектуальных трансляторов.

Практическая ценность состоит в создании комплекса универсальных средств поддержки словарной информационной обработки программных компонент произвольной природы на протяжении всего жизненного цикла программы и его применении в крупноблочном программировании - системе параллельного программирования (СПП) МАЯК штатного системного математического обеспечения многопроцессорного вычислительного комплекса и трансляторе языка системы алгебраического программирования APS. Разработана методика применения словарей программных компонент и развиты элементы информационной технологии в процессе построения программного решения с использованием объектно-ориентированной методологии. Информационная обработка программных компонент в виде словарей и их иерархий обеспечивает сопровождение компонент и поддержку их выполнения, улучшая управление обслуживанием задач и повышая производительность разработки и функционирования программных продуктов.

Реализован многовариантный транслятор процедурного подмножества входного языка системы алгебраического программирования в язык C с оптимизацией, управляемой с помощью иерархических алгебраических спецификаций и осуществляемой на базе иерархии словарей алгебраических модулей и процедур.

Апробация работы и публикации. Научные и практические результаты работы обсуждались на заседаниях республиканского семинара "Теория автоматов и ее применения" (Киев, 1984-1993), Всесоюзного семинара "Параллельное программирование и высокопроизводительные системы" (Уфа, 1990, Планерское, 1991), Всесоюзной конференции "Актуальные проблемы системного программирования" (Таллинн, 1990), Всесоюзной научно-технической конференции "Практическое применение современных технологий программирования, пакетов и прикладных программ" (Днепропетровск, 1990), Международного семинара молодых ученых "New Generation Computer Systems and Software" (Самарканд, 1990), рабочей группы по языкам и системам программирования комиссии СМО АН СССР "Языки, системы и методы программирования для суперкомпьютеров и параллельных ЭВМ" (Ленинград, 1991). Результаты диссертационной работы отражены в 8 статьях, тезисах 2 докладов и в коллективной монографии.

Структура и объем работы. Диссертация состоит из введения, четырех глав, заключения, списка литературы, включающего 167 наименований, и четырех приложений. Объем работы - 145 страниц текста, 16 страниц списка литературы и 58 страниц приложений.

СОДЕРЖАНИЕ ДИССЕРТАЦИОННОЙ РАБОТЫ

Во введении обоснована актуальность темы диссертации, проанализированы основные тенденции и проблемы разработки программного обеспечения для поддержки локкомпонентного программирования, кратко охарактеризована парадигма ООП, основные понятия которой используются в главах диссертации, сформулированы цели и задачи работы, приведено содержание всех разделов диссертации.

В первой главе предлагается аналитический обзор и сравнительный анализ параллельных средств в объектно-ориентированном программировании.

В разделе 1.1 рассмотрен параллельный аспект распределенных вычислений в объектно-ориентированных системах. Описаны три формы параллельности вычислений: внутримодульная, межмодульная и делегирование. Приведена классификация языков, интегрирующих объекты и параллелизм, по способу введения параллелизма в такой язык, на ортогональные, однородные и неоднородные. Исследованы средства параллельности широкого спектра объектно-ориентированных и объектно-базируемых языков: Smalltalk и его разнообразные параллельные разработки (Concurrent Smalltalk, Orient84/K, Distributed Concurrent Smalltalk), языки, поддерживающие акторную модель параллельных вычислений в объектах (ABCL/1, Act1, Act3 и др.), а также такие языки как Emerald, Trellis/Owl, Concurrent C++, POOL, Hybrid, Occam с объектами на транспьютерах, PROCOL и др.

Раздел 1.2 посвящен сравнительному анализу средств интеграции объектов и параллелизма. Описаны преимущества и недостатки трех основных подходов к реализации параллелизма в языках с объектами. При ортогональном подходе хотя возможен теоретически неограниченный параллелизм, может нарушаться целостность объекта. В параллельных языках взаимоисключающее выполнение операций объекта поддерживается автоматически, сохранение целостности объекта происходит за счет ограничений параллелизма и дополнительных затрат.

На основе проведенного исследования выделены характеристики, существенные для систем параллельных распределенных вычислений, базирующихся на объектах:

- организация параллельности: виды параллельности, ее реакция, модель параллелизма, учет "параллельных ролей";
- организация распределенности: единица и масштаб распределенности, организация адресного пространства, возможность миграции объектов в сети;

- соотношение параллельности и распределенности;
- организация взаимодействия: виды взаимодействия, механизмы координации, количество партнеров, режим связи;
- управление вычислениями: управление ресурсами, защита, управление доступом, планирование поведения, расслоение вычислений.

Построена аналитическая таблица координационных средств распределенных вычислений исследованных языков.

Раздел 1.3 посвящен анализу языка параллельного программирования высшего уровня МАЯК, поддерживающего макроконвейерную организацию вычислений, с позиции ООП, и перспективам его развития с использованием объектно-ориентированного подхода.

Дана краткая характеристика языка МАЯК, его свойств и средств: модульность, ориентация на крупноблочную организацию вычислений по данным и по действиям, разнообразия параллельная обработка вычислений, строгая типизация, развитые структуры управления вычислениями, обработка исключительных ситуаций и др.

Основные конструкции языка МАЯК, связанные с организацией параллельных процессов, рассмотрены с точки зрения ООП. Объект языка МАЯК - параллельно функционирующий экземпляр модуля. Основной параллельной программной обработкой является компонента (логический процессор). Главный инструмент для создания различных видов параллельных объектов - описание программного модуля, которое может рассматриваться как задание класса или вида объектов одинаковой структуры. Объект образуется из класса, когда (явно или неявно) происходит связывание этого модуля с компонентой - ресурсом его исполнения. Взаимодействие между объектами-модулями осуществляется с помощью запросов, которые в разных случаях (в зависимости от вида модуля) выражаются теми или иными средствами языка.

Исследованы равнообразные возможности представления объектов в МАЯКе. Инкапсуляция и экспортирование свойств поддерживаются в МАЯКе модулями, внешними типами, внешними процедурами, внешними функциями, внешними массивами, компонентами, сообщениями. Действия над объектом могут быть заданы в операторе обработки сообщения; во внешних подпрограммах, в программах пакета. Внешний интерфейс объекта определяется, соответственно, структурой сообщения, спецификацией внешних подпрограмм, спецификацией программ пакета.

Свойство наследования в МАЯКе определяется отношением подчиненности модулей программы: каждый модуль задает ресурсы, которыми могут воспользоваться все подчиненные ему модули. Наследование

имеет крупномасштабный характер, так как охватывает не только данные и действия, а физические и программные ресурсы. Иерархия классов формируется на основе связей подчиненности модулей в программе и представлена структурой реализации языка - деревом статической подчиненности модулей многомодульной программы МАЯК. Существуют два типа иерархий в МАЯКе: горизонтальная иерархия связей подчинения модулей программы (внутрипрограммное наследование) и вертикальная иерархия функциональной структуризации программных продуктов системы по проблемно-связанным областям - информатекам (межпрограммное наследование).

Подтверждена необходимость развития языка МАЯК на основе явного использования объектно-ориентированной методологии. Сформулированы принципы построения нового поколения языка, как общецелевого координационного языка программирования, сохраняющего семантику языка МАЯК, в котором ОП становится основным способом программирования, на основании чего язык поддерживает технологию генерации эффективных применений. Определен базовый элемент программирования - объект модульного типа, идентифицирующий параллельную единицу распределенных вычислений, в котором унифицированы основные понятия вычислительного процесса.

Во второй главе изложена методология использования информации программных компонент в процессе координации вычислений в многокомпонентной программной среде, и описан комплекс инструментальных средств для автоматической информационной обработки программных компонент, отражающей их организацию и различные аспекты процесса программирования от проектирования до реализации.

В разделе 2.1 введена информационная модель программной компоненты в виде иерархии словарей и определено информационное отображение программной компоненты в словарь.

Словарь - информационный объект, связываемый с некоторой именованной программной областью и содержащий совокупность упорядоченных определений слов - программных элементов ассоциируемой области. Элементом словаря является его статья, представляющая собой пару: слово и его определение. Структура словаря включает заголовок словаря, множество классов словаря и множество характеристик слов словаря; она зависит от специфики обработки программной информации и различна для разных применений. Словарная модель программной компоненты представляет ее онтологию - терминологию, концепцию и парадигму интерпретации.

Информационное отображение программной области определено таким образом, что информация об элементах программной области находится в словаре этой области, размещаясь в его определенном классе так, что каждый класс словаря представляет собой последовательности равноименных слов с характеристиками.

Информационная модель многокомпонентной программы представлена иерархией словарей, в которой отражена компонентная структура программы. Конструкция иерархии словарей моделирует область видимости (действия или среду функционирования) компоненты программы из словарей компонент, которым она подчинена (от которых зависит). На основе иерархии можно проводить разнообразные исследования и управлять средой функционирования программных компонент. Создание словарей для компонент многокомпонентных программ и классификация программных элементов в словаре делают поиск информации, во-первых, корректным, сужая область поиска информации до определенного иерархического подмножества всего множества словарей - области видимости; во-вторых, эффективным, сужая область поиска до некоторого искомого множества объектов - класса(-ов) словаря. Динамическое построение области видимости любой компоненты программы из предварительно созданных словарей всех ее компонент и исследование области видимости обеспечивают независимую трансляцию и раздельное выполнение любой компоненты программы.

Раздел 2.2 посвящен технологии работы со словарями: описаны объектно-ориентированные среда и функциональность словарей.

Способ работы со словарями максимально приближен к естественному: работа с ним осуществляется с помощью интерфейса объекта словарь, состояние которого определяется его структурой, включающей класс (словаря), слово (класса словаря), характеристику слова, а поведение - операциями открыть словарь, закрыть словарь, прочитать словарь, записать словарь, распечатать словарь, записать слово в словарь, записать характеристику слова, найти слово, найти характеристику слова. Объектно-ориентированный метод обработки информации позволяет создавать открытые гибкие словарные информационные системы.

Действия, определяющие функциональность объектов - словарь и иерархия словарей, сгруппированы по способу обработки информации на операции по работе со словарями, по формированию информации в словаре и по использованию информации из словарей иерархии. Технология создания словарей охватывает цикл от создания "пустого"

словаря, заполнения его информацией и до фиксации сформированного словаря в памяти. Технология использования словарей предусматривает (непосредственное или опосредованное другими действиями) расположение словарей в иерархию и включает операции чтения словаря из памяти, поиска информации в словарях, печати словаря. Приведена семантика соответствующих операторов и спецификации реализующих их функций.

Раздел 2.3 посвящен реализации словарного инструментального комплекса. Разработанный комплекс инструментальных средств реализован в расширяющейся системе программирования ТЕРЕМ в виде универсального пакета семантических действий Конструирования и Использования Словарей (КИС), настраиваемого на конкретное применение. КИС - инструментарий для формирования и исследования семантики программной компоненты, обеспечивающий интеллектуальную поддержку процесса программирования. Первая версия словарей была реализована на ЕС ЭВМ и внедрена в СПП МАЯК. Вторая его реализация была осуществлена на ПЭВМ и использована в трансляторе языка системы алгебраического программирования APS.

В разделе 2.4 изложена методика применения словарного инструментария. Рабочий цикл состоит из трех стадий: предварительная программно-текстовая настройка словаря на конкретное применение, создание словарей и их использование. Предусмотрены два режима работы со словарями: специфицируемый, охватывающий минимальный объем стандартной работы по созданию любых словарей, и программируемый, предлагающий широкие возможности по формированию равнообразной информации в специфических словарях и по использованию накопленной информации.

В разделе 2.5 выполнен количественный и качественный анализ обработки информации в виде словарей и их иерархий. Проведены теоретическая оценка поиска информации в словарях и сравнение словарной организации информации с известными табличными методами, такими как последовательные, упорядоченные и таблицы с непосредственным доступом. Словари являются обобщением таблиц символов в трансляторах. Преимущество организации информации в виде словарей состоит в следующем:

- гибкое структурирование равнообразной программной информации, ориентированное на семантику обработки программных элементов в распределенной среде вычислений, а именно, структуризация информации по областям программы (словарям) и по специфике ее обра-

ботки (классам и характеристикам), пользовательская программно-текстовая настройка структуры информации на применение, автоматически: сбор информации в соответствии с выбранной структурой;

- естественный, унифицированный и эффективный способ обработки информации в виде словарей, точнее, однородная обработка основной информации как слов класса, однородная обработка различной дополнительной информации как характеристик слов, текстовая интерпретация закодированной информации, устойчивое хранение словарей, локализация области поиска информации в пределах от области видимости (иерархии словарей) до искомого подмножества информации (класса), конструирование различных стратегий поиска на основе структуризации;
- небольшой дополнительный объем памяти;
- открытость, расширяемость, универсальность системы словарей;
- использование одной информационной структуры для различных аспектов программной обработки на протяжении всего жизненного цикла программы.

В связи с унифицированной, разнообразной и гибкой обработкой информации в виде словарей возможен широкий диапазон применения предлагаемого инструментального комплекса, прежде всего в распределенной архитектуре вычислений. В общем случае он может быть использован в системах моделирования процесса покомпонентного программирования для поддержки информационной обработки программных компонент от проектирования до реализации, например, в мегапрограммировании как средство сопровождения мегамодулей и поддержки выполнения мегапрограмм, при проектировании программ, основанном на знаниях. ИИС может быть использован также в традиционных трансляторах - для статического семантического контроля.

В следующих двух главах описывается применение словарного инструментального комплекса, соответственно, в крупноблочном программировании и трансляторе последовательного языка.

Третья глава посвящена информационной обработке в мегапрограммировании - современной концепции крупноблочного программирования, введенной на Западе в 1990 году для проведения исследований по проблематике крупномасштабного проектирования программ, как инженерии программного обеспечения, основанной на готовых компонентах и управлении жизненным циклом. Проведен анализ технологии макроконвейерного программирования в языке МАЯК с позиции мегапрограммирования.

В разделе 3.1 проведен краткий обзор парадигмы мегапрограммирования: описаны ее отличительные черты, ключевые элементы программной поддержки мегапрограммирования, его программные компоненты - мегамодули и требования к языкам мегапрограммирования.

Раздел 3.2 посвящен анализу системного математического обеспечения многопроцессорного вычислительного комплекса с макроконвейерной организацией вычислений (СМО МК) с позиции мегапрограммирования. Анализ показал, что в процессе реализации микроконвейерного принципа организации вычислений в середине 80-х годов были получены основные компоненты мегапрограммирования: язык, определяющий крупномасштабный компонентно-базированный способ программирования, и инфраструктура, поддерживающая параллельное функционирование сети программных модулей.

Язык МАЯК обеспечивает композицию модулей в виде разнообразных схем координации, поддерживающих различные модели параллелизма. Выделенные в МАЯКе два основных типа модуля - внешний массив - для задания обрабатываемых данных большого объема, и программный модуль - для задания обрабатываемого макродействия, обеспечивают эффективную параллельную обработку данных большого объема. Композиция модулей в координационную схему программы МАЯКа выделяет уровень организации вычислений, обеспечивая гибкое управление конфигурациями компонент, и интегрирует координацию и вычисления, определенные в модулях программы.

Сопровождение и выполнение программ МАЯКа поддерживалось его программным обеспечением: СПП обеспечивала подготовку модулей к исполнению, операционная система - выполнение модулей в разнородном окружении, информ. банки - хранение программных продуктов, функционально структурированных по областям применений, словари - информационную обработку распределенных компонент.

В разделе 3.3 рассмотрена информационная обработка в мегапрограммировании. Описана информационная обработка в макроконвейерном программировании на примере реализации языка МАЯК с помощью предложенного в главе 2 словарного инструментального комплекса.

Проблема координации информации в мегапрограммировании связана с композицией независимых разнородных компонент в мегапрограмме. Для координации разных языков, множественных парадигм программирования необходимо альтернативное информационное представление гетерогенных компонент.

Информационная обработка в мегапрограммировании включает

сопровождение мегамодулей и поддержку выполнения мегапрограмм. Базисом программирования мегамодуля является его онтология, которая позволяет интеллектуализировать обработку модуля и обеспечивает свободное и гибкое управление взаимосвязью модулей. Словарь — один из основных элементов программной поддержки мегапрограммирования, в котором может быть сформирована онтология мегамодуля и на основании которого выполнен его качественный анализ.

Инструментарий КИС обеспечивает единый программный базис для такой информационной обработки. Словарь имеет интерфейс, необходимый и достаточный для формирования онтологии мегамодуля, для задания текстовой интерпретации программной информации, для наблюдения за состоянием модуля как в пассивной — текстово-визуальной форме, так и в активной — программной форме, а также для согласования информации разнородных компонент в процессе выполнения. Для распределенных спецификаций и выполнения мегамодулей словарь может быть размножен и передан в места функционирования модулей.

Информационная обработка в многопроцессорном вычислительном комплексе состоит в предварительном построении среды исполнения многомодульных программ МАЯКА и ее дальнейшем использовании в процессе их функционирования. Информационная среда исполнения программы моделируется в виде словарей ее модулей на этапе декомпозиции многомодульной программы — начальном этапе обработки программ в СПИ МАЯК. Словарь представляет собой онтологию модуля МАЯКА; он создается с помощью пакета КИС на основе внешних описаний модуля в виде расклассифицированной в зависимости от специфики обработки в МВК информации: входная и выходная информация, компоненты, внешние типы, внешние подпрограммы, внешние функции, сообщения, модули, непосредственно подчиненные данному, программы, пакеты. Задается текстовая интерпретация словарной информации. Словари модулей фиксируются в памяти. Использование информации происходит на основе динамически конструируемой иерархии словарей в процессе трансляции модулей программ. Словари модулей обеспечивают возможность распределенного, т.е. раздельного и независимого, выполнения модулей МАЯКА. На основе иерархии словарей реализован также объектно-ориентированный способ функционирования модулей МАЯКА. Класс представлен посредством словаря модуля. Иерархия классов отражена в иерархии словарей. Наследование реализовано как поиск и использование информации из иерархии словарей, на соответствующих словарных разделах.

В четвертой главе рассматривается транслятор процедурного подмножества входного языка APLAN системы алгебраического программирования APS в язык С. Транслятор включает словарный комплекс в качестве ключевой компоненты.

В разделе 4.1 введены основные понятия системы алгебраического программирования: алгебра термов, множество первичных объектов и сигнатура ее операций, алгебраические программы и модули, алгебраические объекты, язык алгебраического программирования APLAN, процедуры языка APLAN. Обоснована необходимость трансляции процедурного подмножества интерпретируемого нетипизированного языка APLAN в язык С для более эффективной работы в системе.

В разделе 4.2 сформулирована задача трансляции с оптимизацией языка L2B - процедурного подмножества языка APLAN, в язык L2C - расширение языка С для обработки системных структур данных. Перечислены средства оптимизации выходного С-кода алгебраических выражений L2B-процедур APLAN-модуля, основным источником и средством которой являются вводимые пользователем иерархические алгебраические спецификации типов и оптимизированных операций. Сформулировано правило перевода алгебраического выражения, согласно которому транслятор гарантирует перевод, соответствующий общей семантике APLANa, и осуществляет оптимизированный перевод алгебраического выражения, если для всех его отметок заданы оптимизированные операции, и типы, специфицируемые и реальные, совпадают.

В разделе 4.3 описаны расширение и ограничения APLANa в связи с трансляцией процедур. Иерархические алгебраические спецификации вводятся с помощью специальных атомов и имен APLANa и образуют расширенный APLAN. Трансляция L2B-процедур с оптимизацией накладывает некоторые ограничения на имена целого типа, на алгебраические программы, на транслируемые процедуры.

В разделе 4.4 формально описаны этапы трансляции: предварительный этап построения среды трансляции, этап анализа, этап трансляции и этап модификации входного файла. Они определяют архитектуру транслятора, которая состоит из следующих компонент:

- словари алгебраических модулей и процедур - статические информационные структуры,
- матрица типов выражения - динамическая информационная структура, формируемая транслятором для каждого выражения,
- множество перзодов операторов и выражений, которое включает общий и оптимизированный варианты их трансляции.

Базовыми действиями транслятора являются: построение словарей, построение матрицы типов выражения на базе словарной информации, контроль типов выражения на основе матрицы типов, статический семантический контроль на основе словарей, выбор способа перевода и перевод.

Трансляция языка L2B в язык С производится на основе статической и динамической информационной среды, которые рассмотрены в разделе 4.5.

Статическая информационная среда трансляции представлена в виде словарей. Предметная область трансляции формально описана в виде множества первичных объектов и их разбиений, производимых под действием спецификаций оптимизации. Структура информационной среды трансляции представляет собой иерархию словарей, состоящей из словаря глобальной информации и словаря локальной информации. Структура словаря отражает описанное разбиение множества первичных объектов и учитывает выполняемый в процессе дальнейшей обработки семантический анализ. Классы словаря алгебраических программных компонент следующие: алгебраические объекты, глобальные целые, транслируемые процедуры, формальные параметры, локальные алгебраические объекты, локальные целые, отметки, бинарные отметки, внутренние процедуры. Информационная среда формируется и используется с помощью словарного инструментального комплекса КИС. Введен функции анализа, в которых непосредственно используется словарная информация и которые участвуют в формальном описании трансляции (раздел 4.6).

Динамическая информационная среда представлена в виде матрицы типов выражения, входящего в оператор. Ее необходимость обусловлена неоднозначностью трансляции алгебраических выражений. Для определения способа перевода для каждого выражения формируется матрица типов на основе статической информации словарей. Матрица типов - структура данных, в которой заданы разные проекции типов обрабатываемого выражения: проекции на его внешние условия или контекст выражения, на язык С и на язык APLAN. В зависимости от их комбинации (соответствия выражения его контекстным условиям) выбирается общий или оптимизированный способ перевода алгебраического выражения. Введен и записана в виде системы соотношений функция перевода выражения.

В разделе 4.6 трансляция формально описана в виде основной системы соотношений `trans()` и нескольких вспомогательных. Введены

языки описания: входной язык L2B, выходной язык L2C, язык транслятора LTR. Правила систем соотношений состоят из условий — анализа предметной области, левой части — транслируемой конструкции входного языка и правой части — результата трансляции. В основной системе соотношений trans() представлены переводы всех L2B-операторов. Во вспомогательных системах соотношений описаны общий и специальный способы трансляции выражения, а также оптимизированный способ трансляции выражений целого типа и др.

В разделе 4.7 описаны особенности реализации транслятора с помощью системы Терем. Приведена аргументация выбора средства реализации, обеспечивающего неоднородную обработку входного языка, динамическую модификацию грамматики, управление количеством просмотров выражений, выбор варианта перевода из множества переводов оператора.

В заключении приведены основные результаты работы и перспективы ее развития.

В приложениях приведены: аналитическая таблица координатных средств языков программирования, интегрирующих объекты и параллелизм, инструментальный комплекс программ информационной обработки программных компонент в виде словарей и их иерархий, структура словаря модуля многомодульной программы МАЯК, описание реализации и спецификация транслятора L2B-L2C, набор тестовых алгебраических программ и протокол работы транслятора с демонстрацией словарей алгебраических программ и процедур; документ, подтверждающий использование результатов диссертационной работы.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

В работе получены следующие основные результаты.

1. Проведен классификационный аналитический обзор параллельных средств широкого спектра объектно-ориентированных и -базируемых языков программирования. Выделены основные характеристики параллельных распределенных вычислений, базирующихся на объектах, и с учетом их построена аналитическая таблица языковых координатных средств распределенных вычислений. Описана модель параллельно функционирующей объектно-ориентированной программной компоненты.

2. Проведен анализ технологии макроконвейерного программирования в языке МАЯК и его среды с учетом многопроцессорной архитектуры инструментальной ЭВМ с позиций объектно-ориентированного программирования и мегапрограммирования. Показано, что объектная

ориентация неявно присутствует в языке, непротиворечиво интегрирована с параллельными средствами и имеет универсальный характер. МАЯК обладает достаточными языковыми средствами для представления концепции мегапрограммирования. Для превращения его в общецелевой координационный язык необходимо введение дополнительных средств с явным использованием понятия объекта. Сформулированы принципы нового поколения языка МАЯК и разработан общий вид базового элемента программирования в языке.

3. Построена информационная модель программной компоненты распределенных вычислений в виде иерархии словарей. Разработан и реализован словарный инструментальный комплекс поддержки информационной обработки программных компонент. Разработана технология применения словарей в процессе покомпонентного программирования, в частности, в мегапрограммировании для поддержки сопровождения и выполнения программ и программных компонент. В системе параллельного программирования МАЯК разработан и реализован пакет информационной обработки модулей многомодульной программы для их независимой трансляции и объектно-ориентированного функционирования.

4. Разработан и формально описан проект транслятора L2B-L2C процедурного подмножества интерпретируемого нетранслируемого языка APLAN системы алгебраического программирования APS в язык C с оптимизацией, управляемой с помощью иерархических алгебраических спецификаций.

5. На основании проекта реализован многовариантный транслятор L2B-L2C с оптимизацией, осуществляемой с помощью информационной обработки на базе иерархии словарей алгебраических модулей и процедур.

Основные положения диссертации опубликованы в следующих работах:

1. Валькевич Т.А., Дудко Е.А., Мищенко Н.М. Декомпозиция мультимодульных программ // Организация вычислений в многопроцессорных ЭЕМ. - Киев:Ин-т кибернетики им. В.Л.Глушкова АН УССР, 1985. - С. 3-11.

2. Валькевич Т.А. Словари мультимодульных программ // Проектирование многопроцессорных вычислительных систем. - Киев:Ин-т кибернетики им. В.М.Глушкова АН УССР, 1986. - С. 37-44.

3. Системное математическое обеспечение многопроцессорного вычислительного комплекса ЕС / Л.С. Михалевич, Т.А. Валькевич,

Ю. В. Капитонова, А. А. Летичевский и др. — М. : ВВИА им. проф. Н. Е. Жуковского, 1986. — 390 с.

4. Валькевич Т. А. Объектно-ориентированный подход к построению словарей мультимодульных программ // Всесоюз. конф. «Актуальные проблемы системного программирования. Объектно-ориентированное программирование»: Тез. докл. (Таллинн, 16—18 янв. 1990). — Таллинн: Ин-т кибернетики АН Эстонии, 1990. — С. 76—77.

5. Валькевич Т. А., Мищенко Н. М., Федурко В. В., Шеголева Н. Н. О внутреннем представлении мультимодульных программ в МВК // X Всесоюз. семинар «Параллельное программирование и высокопроизводительные системы. Методы представления знаний в информационных технологиях»: Тез. докл. (Уфа, 19—26 июня 1990). — Киев : Ин-т кибернетики им. В. М. Глушкова АН УССР, 1990. — С. 156—157.

6. Валькевич Т. А. Пакет процедур для обработки словарей мультимодульных программ // II Всесоюз. науч.-техн. конф. «Практическое применение современных технологий программирования, пакетов и прикладных программ в вычислительных системах и сетях ЭВМ»: Тез. докл. (Днепропетровск, 19—20 сент. 1990). — Днепропетровск: НПО «Орбита», 1990. — С. 86—88.

7. Валькевич Т. А. Объектно-ориентированный подход к работе с таблицами символов // Проблемы создания интеллектуальных систем поддержки принятия решений. — Киев : Ин-т кибернетики им. В. М. Глушкова АН УССР, 1990. — С. 51—55.

8. Валькевич Т. А. Словари мультимодульных программ и их применение в языковых процессорах // УСнМ. — 1991. — № 2. — С. 51—59.

9. Валькевич Т. А. О моделировании среды функционирования объектов в языке МАЯК // Методы и средства объектно-ориентированного программирования. — Киев : Ин-т кибернетики им. В. М. Глушкова АН УССР, 1991. — С. 7—13.

10. Валькевич Т. А. Об объектной ориентации в МАЯКе // Кибернетика. — 1993. — № 2. — С. 159—165.

11. Валькевич Т. А., Гороховский С. С. Языки программирования с объектами и параллелизмом: анализ и характеристики. — Киев, 1993. — 19 с. — (Препр. / АН Украины. Ин-т кибернетики им. В. М. Глушкова; 93-27).

ІНБ ім. В. Стефаніка
АН України

Подп. в печ. 15.06.94. Формат 60×84/16. Бум. тип. № 2. Офс. печ. Усл. печ. л. 1,05. Усл. кр.-отт. 1,17. Уч.-изд. л. 1,5. Тираж 100 экз. Заказ 706.

Редакционно-издательский отдел с полиграфическим участком
Института кибернетики имени В. М. Глушкова НАН Украины
252650 Киев ГСП 22, проспект Академика Глушкова, 40

AB 30.627

AB 30.627