ТЕОРИЯ И ПРАКТИКА ПРОГРАММИРОВАНИЯ И МОДЕЛИРОВАНИЯ ЗАДАЧ ПРЕДМЕТНЫХ ОБЛАСТЕЙ ЗНАНИЙ

К 100-ЛЕТИЮ АКАДЕМИКА

ВИКТОРА МИХАЙЛОВИЧА

ГЛУШКОВА

УДК 001 18 004
ББК 32.81
С32

Автор книги Лаврищева Екатерина Михайловна – доктор физико-математических наук , почетный профессор МФТИ, завотделом ИПС

Лаврищева Е.М. Искусственный интеллект, теория и практика программирования и моделирования программ задач ИИ и предметных областей знаний. Перспектива развития сборки для ИИ и систем представления знанийц
-Москва, 2023.- 201с.

ISBN 9768-966-360-110-6

Посвяыщается академіку Віктору Михайловичу Глушкову в честь його 100-летия з дня народження (23.08.1923)

ВСТУП

Выходили книги к 80-летию и к 90-летию академика Глушкова [1, 2] его учеников и коллег, в которых сформульованы и определены его парадигмы, работы по искусственному интеллекту, информатике, технологии программирования физических, математических и медицинских задач; макропроцессорным и макроконвейерным отечественных компьютерам и организации вычисления на них разных задач научного, физического, технического, хозяйственного, экономического и промышленного назначения. В стране были созданы отечественные компьютеры МЭСМ, М-20, БЭСМ, ОС ЕС ЭВМ, Мир-1-3, Днеп-1, 2 и ЭВМ специального военного назначения с системным программным обеспечением и операционной системой функционирования при обработки задач на них. Впервые в нашей стране сформировалось наряду с ООП сборочное программирования АПРОП технических устройств, прикладных программных и информационных (интеллектуальных) комплексов (ПРОМЕТЕЙ, ПРОТВА, РУЗА и др.) Военного назначения (1967-1980). Под его управления была создана на Днепр-2 в ГДР АСУ ТП металургической промышленности на примере медицинских приборов (Дрезден- - Берлин), АСУ, АСУ ТП, ОГАС) і програмних систем реалізації науково-технічних задач на ЕОМ. Глушков був ідеологом поступового переходу від ремісничого випуску комп'ютерів і програм до промислового їх виготовлення для масового застосування. Він зазначав, що промисловий їх випуск має ґрунтуватися на технологічних лініях конвеєрного їх виготовлення, як це створюється на автомобільних конвеєрах Форда. Технологію програмування В.М.Глушков вважав головною силою прогресу фундаментальних кібернетичних і комп'ютерних наук, спрямованої на створення нових ЕОМ, їх системного забезпечення з вдосконаленням схемної інтерпретації мови ЕОМ до рівня мов програмування, побудови мереж обчислювальних центрів (ОГАС) і прикладного математичного забезпечення автоматизованих систем (АС). Фактично ідеям і концепціям технології програмування відповідав розвиток Computer Science. І коли у 1968 р. на Європейській конференції НАТО пролунав термін – Software Engineering (SE) під лозунгом – продуктивність, якість і індустрія, Глушков на конгресі ІФІП (1972) обгрунтував перспективний шлях збиральної індустрії комп'ютерів і програм.

Нині технології програмування динамічно розвиваються від програмування й реалізації задач проектування обчислювальних систем до побудови великих і малих комп'ютерних фреймворків,

систем та кластерів. Огляд динаміки розвитку технологій програмування ми почнемо з періоду виникнення комп'ютерних й інформаційних технологій і програмних систем, простежимо етапи вдосконалення індустрії програмних продуктів завдяки інтеграції готових ресурсів і збиранню їх на фабриках програм, а також приділимо увагу сучасним методикам викладання курсів з технології програмування для студентів вищих навчальних закладів.

В книге описан творческий путь академика В.М. Глушкова по развитию искусственного интеллекта (ИИ) при решении математических задач с доказательством правильности, автоматического распознования символов в текстах и машинным переводов На первой конференции по искуственному интеллекту были доклады по многим аспектам машиного перевода, построения формальных моделей и обучающихся систем. Описывается приезд в Киев Винера – первого создателя компьютера и построения интеллектуальных систем. В.М. Глушков поддержал идею Винера и начал развивать науку ИИ. Приводятся работы учеников и коллег В.М. Глушкова по ИИ, алгоритмам обучения, задания изображений, абстрактным автоматам, дискретным автоматам и др. Работы проводились на первых ЭВМ \ Киев, М-20, БЭСМ, Мир, Терем и др.

Приводится концепция В.М.Глушкова по прогаммирования математических, Физических задач в ЯП и реализация програмных объектов (модулей, компонентов, сервисов) и их сблорки в системы решения задач.

Прикладных облачтей знаний. Он первый высказал концепцию сборки бортовых систем в авиации, космосе, морфлоте , автомобильной и другой промышленности, а также построения АСУ ТП для разных областей знаний , промышленности и др. С помощью системы автоматизации программ –АПРОП, сделанной под руководством В.М.Глушкова в рамках НИЦЕВТ (1975-1976) и ВПК (1977-1989) с участием Липаева В.В. (АРГОП, ПРОМЕТЕЙ, И др.)..

УДК 001.18ю004
ББК 32ю81

Введение

Согласно постановления Совета Министров УССР от 28 ноября 1957 года и Президиума АН УССР 13 декабря 1957 года на базе Лаборатории №2 (Вычислительной математики и вычислительная техника) Института математики АН УССР был образован Вычислительный центр АН УССР (ВЦ) на правах научно-исследовательского института. Директором ВЦ был назначен молодой доктор физико-математических наук Глушков Виктор Михайлович, который возглавлял Лабораторию №2 с июля 1956 года. В первой половине 1958 года в ВЦ были введены в действие две универсальные вычислительные машины – "Урал" и "Киев".

Глушков В.М. к тому времени был уже известным специалистом в наиболее абстрактной области искусственного интеллекта, технологии программирования, математической науки алгебры ЭВМ в качестве инструмента для вычисления математических задач, но и помощником человека в других сферах его интеллектуальной деятельности – то есть, быть воплощением искусственного интеллекта на ЭВМ. С первых дней существования ВЦ В.М. Глушков считал приоритетным заданием расширение сферы применения ЭВМ в различных направлениях человеческой деятельности.Ряд научных разработок Академика В.М.Глукшкова представлены в книгах, посвященных в 80-и 90летию Глушкова. В книге к 100 летию глушкова Е.М. Лаврищева приводит работы В.М. в Области ИИ и технологии программирования.

Р а з д е л 1

Научные идеи В.М. Глушкова по искусственному интеллекту и реализации ИИ на первых ЭВМ

Мищенко Н.М., Лаврищева Е.М.

Аннотация. Описан творческий путь академика В.М. Глушкова по развитию искусственного интеллекта (ИИ) при решении математических задач с доказательством правильности, автоматического распознования символов в текстах и машинным переводов На первой конференции по искуственному интеллекту были доклады по многим аспектам машиного перевода, построения формальных моделей и обучающихся систем. Описывается приезд в Киев Винера – первого создателя компьютера и построения интеллектуальных систем. В.М. Глушков поддержал идею Винера и начал развивать науку ИИ. Приводятся работы учеников и коллег В.М. Глушкова по ИИ, алгоритмам обучения, задания изображений, абстрактным автоматам, дискретным автоматам и др. Работы проводились на первых ЭВМ Киев, М-20, БЭСМ, Мир, Терем и др.

Введение

Согласно постановления Совета Министров УССР от 28 ноября 1957 года и Президиума АН УССР 13 декабря 1957 года на базе Лаборатории №2 (Вычислительной математики и вычислительная техника) Института математики АН УССР был образован Вычислительный центр АН УССР (ВЦ) на правах научно-исследовательского института. Директором ВЦ был назначен молодой доктор физико-математических наук Глушков Виктор Михайлович, который возглавлял Лабораторию №2 с июля 1956 года. В первой половине 1958 года в ВЦ были введены в действие две универсальные вычислительные машины – "Урал" и "Киев".

Глушков В.М. к тому времени был уже известным специалистом в наиболее абстрактной области математической науки алгебры и сразу оценил возможности ЭВМ в качестве инструмента для вычисления математических задач, но и помощником человека в других сферах его интеллектуальной деятельности – то есть, быть воплощением искусственного интеллекта на ЭВМ. С первых дней существования ВЦ В.М. Глушков считал приоритетным заданием расширение сферы применения ЭВМ в различных направлениях человеческой деятельности.

1. Основные направления развития ИИ Глушковым В.М.

В начале 1958г. В.М.Глушков использовал *универсальную электронную вычислительную машину "Урал" для проверки правильности доказательства теорем в алгебраической теории.* Приоритет развития искусственного интеллекта в деятельности В.М. Глушкова подтвердился и во время его визита в США в составе советской делегации в апреле-мае 1959 года, состоявшегося в рамках обмена делегациями ученых между США и Советским Союзом. Среди заявленных им в анкете интересов был и такой, *machine learning* ([2, с.491]). Этот термин многогранный, одну из его граней можно коротко охарактеризовать как дисциплину, суть которой состоит в разработке компьютерных программ, способных самосовершенствоваться в процессе выполнения на компьютере.

В это же время В.М. инициировал исследования в таких направлениях как автоматическое распознавание символов и изображений, лингвистические исследования (машинный перевод, статистическая обработка текстов), моделирование биологической эволюции, обучение машины распознавать смысл простых фраз на естественном языке и др. Параллельно с работами в области искусственного интеллекта и решения математических задач на ЭВМ в ВЦ АН Украины начали вестись интенсивные работа по внедрению вычислительной техники в народное хозяйство, в частности, в управление технологическими процессами. В.М. много усилий приложил к тому, чтобы применить вычислительную технику для управления экономикой страны, – впервые в мировой практике образовал общегосударственную компьютерную информационную модель экономики СССР. И достиг неплохих результатов, несмотря на большое сопротивление этим новшествам со стороны старой управленческой команды СССР.

6-9 июня 1960 года в ВЦ состоялась Вторая научная конференция по вычислительной математике и вычислительной технике. Тематика докладов, представленных в тезисах [3], позволяла ознакомиться со всем спектром работ, выполняемых ВЦ в то время. В этом состоит определенная историческая ценность тезисов – ведь с тех пор прошло более 60 лет. "Иных уж нет, а те далече" – можно сказать и о сотрудниках ВЦ 1960 года.

На конференции были представлены 96 докладов, из них об управляющей машине широкого назначения 14 докладов, состоявшихся после доклада Б.Н. Малиновского "*Использование цифровых вычислительных машин для автоматизации промышленности*". 6 докладов было посвящено полупроводниковым устройствам, по нескольку докладов в каждом из таких направлений: применение компьютеров в различных отраслях промышленности; усовершенствование и эффективное использование ЭВМ МЭСМ, "Киев", СЭСМ, "Урал"; автоматизация программирования; программирование методов вычислительной математики.

К теме искусственного интеллекта отнесены были такие доклады:

Н.М. Грищенко, Т.А. Крайнова, С.Н. Якименко. *О программировании алгоритма независимого анализа языка.*

Л.А. Калужнин, А.А. Стогний, Л.С. Стойкова. *Математические принципы построения автоматического словаря для машинного перевода.*

Л..А. Калужнин, С.Н. Якименко. *Статистическое исследование печатных текстов.*

В.А. Ковалевский, А.Г. Семеновский. *Автоматическое распознавание букв и цифр методом анализа.*

И.П. Севбо. *Статистика флексий на материале русского языка 1-ого уровня.*

Э.Ф. Скороходько. *Один способ построения формальной модели значения и некоторые возможности его применения.*

А.А. Стогний. *О принципах построения одной обучающейся программы.*

2. 1-Международный конгресс IFAC (International Federation of Automatic Control) 27 июня - 7 июля 1960 г. в Москве

На конгрессе участвовало более 150 зарубежных делегатов. Это была едва ли не первая многолюдная международная научная акция по кибернетике в Советском Союзе. В американском журнале [4, с.759-776] была напечатана статья под названием "Soviet Cybernetics and Computer Sciences – 1960", в которой американский делегат конгресса Edward A. Feigenbaum, представил информацию о конгрессе, а также описал встречи с советскими учеными-кибернетиками, в том числе с В.М. Глушковым. В частности статья свидетельствовала о том, что 28 июня в Политехническом музее в Москве выступал делегат конгресса Норберт Винер. Зал был полон (500-700 человек). Он говорил о мозговых волнах (brain waves) и о результатах своих электроэнцефалографических исследований. Большинство вопросов касались сравнения творческих возможностей мозга человека и компьютера. Больше всего аплодисментов вызвало

высказанное Н. Винером убеждение в том, что творчество человека всегда будет превосходить машинное.

Н.Винер прибыл в Киеве и выступал в Киевском доме научно-технической пропаганды. Его интерес к Киеву объяснялся еще и тем, что он родился в Украине, и в то время еще были живы его родственники, с которыми ему так и не разрешили встретиться. Его доклад слушали в переполненном зале КДНТП.

Делегат конгресса Edward A. Feigenbaum имел возможность после доклада посетить несколько городов Советского Союза. Он знал о Киеве как о главном центре кибернетических исследований в СССР. Запланировал срок – два дня в Киеве, который независимо от него был сокращен на один день. Он позвонил в ВЦ из аэропорта – и был немедленно доставлен машиной, а затем тепло встречен В.М. Глушковым в ВЦ. В.М. расказал гостю о своей работе по теории абстрактных автоматов. Рассказал также о проблеме обучения компьютера распознавать осмысленность фраз естественного языка, над которой работали его аспиранты. Приводем почти дословный перевод с английского сказанного В.М. делегату конгресса о проблеме, история решения которой приводится ниже.

...Пусть выбран ограниченный словарь, который состоит из 20 существительных имен, 15 глаголов и 15-20 предлогов. Как только эти слова выбраны, можно строить все осмысленные предложения, содержащие эти слова, и затем ввести их в машину. Но это пробный путь, с учетом большого количество возможных вариантов предложений.

Цель состояла в том, чтобы задать машине лишь определенное число осмысленных предложений и после их обработки попросить машину ответить, осмысленно или нет новозаданное предложение. Если машина ошиблась, то указать на ошибку. Идея состоит в том, чтобы машина сформировала классы из тех осмысленных предложений, которые были уже поданы. Например, пусть сформирован класс объектов, которые могут стоять – дом, мальчик, человек, ребенок. Если предложение с глаголом "думать" будет подано впервые, машина правильно ответит, что "мальчик думает", "мужчина думает", "ребенок думает", а "дом думает" – ошибка. Тогда она должна сформировать новый класс обьектов, которые "стоят, но не думают". Число классов может быть очень большим, но намного меньше множества всех осмысленных предложений, которые можно построить со слов данного словаря. Этот процесс расщепления и формирования классов можно инициировать и для предложений "имя существительное-глагол-предлог-имя существительное".

В.М. Глушков достаточно детально рассказал делегату о работах в ВЦ по распознаванию букв. Его рассказ делегату конгресса об алгоритме обучения машины распознавать простые осмысленные предложения был первым из найденных печатных свидетельств В.М. об этой теме.

В заключение статьи делегат конгресса представил свое видение состояния компьютерных наук в Советском Союзе. Он был согласен с учеными из США, посетившими СССР, что США имеет определенные преимущества в проектировании и изготовлении компьютеров, но нет разрыва в фундаментальных идеях – возможно, лишь за исключением изготовления надежных транзисторов.

В отличие от Н. Винера В.М. Глушков был оптимистом в отношении развития искусственного интеллекта в будущем. В "Литературной газете" за 1.01.1976 года была опубликована беседа В.М. с журналистом В. Моевым, который поставил В.М. такой вопрос:

"*Оставим пока в стороне споры, как сложатся взаимоотношения естественного и искусственного интеллекта, но хотел бы услышать четко: можно ли вообще создать искусственный разум, равный человеческому*?" .

Ответ В.М.: "*Наверное, ответом должна послужить вся беседа, я постараюсь изложить, обосновать свою точку зрения. Но если уж вам так важно услышать это в самом начале, извольте – "да" и еще раз "да". ... Можно ли создать полноценный искусственный разум – уже не вопрос. Безусловно, можно. Причем человеческому он не только не уступит, но во всех*

отношениях опередит его. Это произойдет, видимо, еще до начала XXI века... У нас в Институте кибернетики программа научно-исследовательских работ по этой теме принята и рассчитана примерно на такую перспективу..."

Дальше идет детальное изложение и обоснование позиции В.М….

К вопросу, может ли машина мыслить, относится и алгоритм обучения машины распознавать осмысленные предложения естественного языка. Этому алгоритму В.М. придавал большое значение, о чем свидетельствуют его описания в нескольких изданиях, например, [1], [5], [6].

С введением в эксплуатацию ЭВМ "Киев" в 1959 году появилась возможность реализовать программно некоторые алгоритмы искусственного интеллекта, в том числе и вышеупомянутый алгоритм В.М. Полный перечень программ для ЭВМ "Киев", созданных до 1962 года, имеется в монографии [7].

3. Работы сотрудников отдела В.М. Глушкова в области ИИ

Ст. инженер Иваненко Л.Н. был автором трех программ для ЭВМ "Киев". Две из них – в отделе программирования в 1960 году по алгоритмам: морфологического анализа словоформ на примере русского языка (алгоритма И.О. Мельчук, Институт языкознания АН СССР, Москва) и арифметического блока программирующей программы-2 (ПП-2).

Через полгода после окончания Киевского госуниверситета им. Т. Шевченко (июнь 1960 года) Иваненко Л.Н. был переведена из отдела программирования в отдел В.М. Глушкова, где в начале 1961 года начала работать над составлением своей третьей в жизни программы для ЭВМ "Киев" по алгоритму В.М. Глушкова – программы обучения ЭВМ распознавать осмысленные предложения русского языка, совершенствуя процесс распознавания на основании опыта. 1961 год был годом программирования и проведения экспериментов на машине "Киев". Ниже приводится постановка задачи из монографии [1, сс.152-157].

... Весьма важную область применения самосовершенствующихся систем алгоритмов составляют задачи обучения языку.... Речь идет о задаче обучения распознаванию смысла фраз на том или ином языке. Реально рассматривался русский язык и фразы весьма простой грамматической конструкции с достаточно бедным словарным запасом (порядка 100 слов) (по техническим причинам – Н.М.). *В настоящем изложении мы ограничимся лишь основными идеями, которые были положены в основу алгоритма обучения распознаванию смысла.*

Предположим сначала, что речь идет о распознавании смысла фраз, состоящих только из подлежащих и сказуемых. Запас слов, из которых составляются изучаемые фразы, заранее фиксируется. Ясно, что при этих условиях возможно лишь конечное число фраз, и тем более, среди них конечное число осмысленных. Понятие "осмысленности" фразы точно не определяется. Предполагается просто, что "учитель" каким-то способом, придерживаясь, однако, обычного житейского понятия о смысле или бессмысленности, произвел разбиение всех фраз на два непересекающиеся класса: класс осмысленных фраз и класс бессмысленных фраз. В процессе обучения каждой фразе, подаваемой на вход алгоритма, сопоставляется признак ее принадлежности одному из этих двух классов....

Точная постановка задачи об обучении распознаванию смысла фраз состоит в следующем: необходимо построить самосовершенствующуюся систему алгоритмов, которая после сообщения ей некоторого числа N1 случайно выбираемых фраз из общего числа N осмысленных фраз данной конструкции научилась бы правильно распознавать осмысленность любой фразы этой же конструкции, то-есть, относить эту фразу либо к числу осмысленных, либо к числу бессмысленных фраз...

В процессе обучения различают два режима: режим обучения и режим экзамена. В режиме обучения системе подают некоторое число (не все) осмысленных фраз. Затем в режиме экзамена подают фразы, а система должна отвечать, они имеют смысл или бессмысленны. В случае

неправильного ответа можно делать подсказку и таким образом обучать в режиме экзамена. Если в режиме обучения подать все осмысленные фразы, то машина всегда будет отвечать правильно.Такой процесс ("зубрежка") называется тривиальным.

Заслуживает внимания нетривиальный процесс, когда системе подается часть осмысленных фраз, а она начинает правильно отвечать на все поданные ей во время экзамена осмысленные фразы. Возможность такого рода обучения основывается на существовании связей между осмысленными фразами, позволяющими на основании осмысленности одной фразы делать правдоподобные заключения об осмысленности некоторых других фраз. Связь такого рода устанавливается, например, высказыванием, что "почти все думающие являются вместе с тем ходящими".

Предлагаемая самосовершенствующаяся система для распознавания смысла фраз, которую мы будем называть смысловым дискриминатором, основывается на идее фиксации связей между различными осмысленными фразами посредством введения новых понятий. Применительно к фразам простейшей конструкции "подлежащее – сказуемое" целесообразно вводить новые слова (понятия) для обозначения классов существительных, сочетаемых с теми или иными множествами глаголов.

Первоначально такие классы образуются из существительных, которые сочетаются с одним глаголом. Получаем, например, классы "думающих", "говорящих" и т.п. После образования некоторого числа одноглагольных классов может оказаться целесообразным вводить многоглагольные классы. Пусть, например, в исходном словаре были существительные "студент", "профессор", "отец", "трактор" и глаголы "работать" и "думать". Пусть во время обучения случайно поступили фразы с этими существительными и только с глаголом "работать". Этот глагол объединил их в один класс "работающих". После этого на вход дискриминатора поступили несколько осмысленных фраз с глаголом "думать". Представилось естественным объединить класс "думающих" с классом "работающих". При таком объединении возможны ошибки, например, "трактор" оказался в классе "думающих". Такие ошибки могут выявиться в процессе экзамена, и тогда "трактор" будет выведен из класса "думающих".

Таким образом, смысловой дискриминатор в процессе обучения фактически создает новое понятие, не входящее в первоначально заданный ему словарный запас. Благодаря этому создается возможность более экономного задания существующих в языке смысловых связей по сравнению с простым запоминанием всех осмысленных фраз.

Именно в таком виде смысловой дискриминатор и был запрограммирован. Однако, в отличие от описанного выше, в запрограммированном алгоритме разрешалась конструкция фраз типа *подлежащее-сказуемое-дополнение* (с предлогом или без). Программа состояла из 400 команд ЭВМ "Киев". В архиве программиста сохранились данные лишь об одном эксперименте (40 имен существительных, 51 глагол и несколько предлогов), во время которого сформировано 12 классов имен существительных. Названия классов – понятия: *люди*, *посуда*, *мебель* и др. Основная работа по усовершенствованию программы и постановке экспериментов длилась до середины 1962 года, когда программист получила новое задание.

Во время экспериментов был обнаружен один курьезный факт. После обучения машины ей была подана для проверки на осмысленность нехитрая фраза: *инженер находится в кухне*. Машина оценила ее как бессмысленную. Ошибки в программе найти не удалось. Осталось "поверить" машине. Этот факт всем понравился и даже попал в прессу. Так, в газете "Неделя" от 23-30 ноября 1983 года в статье о В.М. Глушкове упоминается весьма успешная попытка научить машину распознавать осмысленность простых фраз, и в то же время машина не захотела признать фразу о присутствии инженера в кухне как имеющую смысл. Так и осталась метка на совести программиста.

3.1. Всесоюзный симпозиум "*Принципы построения самообучающихся систем*" в Киеве 5-6 мая 1961 года

Доклад "*Об обучении распознаванию осмысленных предложений на ЭЦМ*".

А.А. Стогний и Н.М. Грищенко.

В первый день работы симпозиума В.М. Глушков сделал доклад с новыми фактами усовершенствования алгоритма и о конкретных результатах обработки текстов и перспективах исследованиях.

Кроме доклада В.М. Глушкова в программе были объявлены еще два доклада с ВЦ по искусственному интеллекту:

В.М. Глушков, В.А. Ковалевский, В.И. Рыбак. "*Об одном алгоритме обучения распознаванию образов*" (на простых геометрических фигурах).

А.А.Летичевский, А.А. Дородницына. "*Моделирование естественного отбора*". Эксперимент демонстрировался в ВЦ на дисплее, присоединенном к машине "Киев".

В программе было представлены 14 докладов. Среди них было два интересных из Москвы:

А.А.Ляпунов, Ю.Ю. Финкельштейн (Москва). *О формировании поведения коллектива автоматов*

О.С. Кулагина (Москва). *О выработке алгоритмов машинного перевода при помощи ЭЦМ.*

Некоторые доклады симпозиума были напечатаны в 1962 году в сборнике [8]. В нем нет доклада О.С. Кулагиной, но есть несколько статей в программе симпозиума:

А.А. Стогний. *Некоторые математические вопросы построения цифровой логической машины*

В.М. Глушков, В.А. Ковалевский, В.И. Рыбак. *Универсальная установка для исследования алгоритмов распознавания изображений*

А.И. Кухтенко. *О некоторых классах самонастраивающихся систем автоматического управления*

А.Г. Ивахненко. *Индуктивный и дедуктивный методы познания как основа построения двух основных типов обучающихся систем*.

3.2. 4-й Всесоюзный математический съезд 12 июля 1961 года в Ленинграде - Грандиозное событие.

На этот съезд с ВЦ было командировано несколько математиков и программистов. Все летели в одном самолете с В.М. Глушковым. В пути он показывал через иллюминатор разные выдающиеся места, озера Чудское, Ладожское. Сотрудники ВЦ бегали от одного иллюминатора к другому. Стюардессы волновались и делали замечания, но на них не обращали внимания.

Программа съезда была очень насыщенной. 31 пленарный доклад, из которых по 4 часовых доклада каждый день произносились одновременно в разных местах города. Следовательно, на протяжении дня можно было слушать только один пленарный доклад. Напечатаны аннотации пленарных докладов в среднем по полстраницы каждая. Приведем названия пленарных докладов, касающихся кибернетики:

Глушков В.М. *Алгебраическая теория автоматов*

Колмогоров А.Н. *Дискретные автоматы и конечные алгоритмы*

Новиков П.С. *Теория алгоритмов и вопросы алгебры*

Шура-Бура М.Р., Ершов А.П. *Машинные языки и автоматическое программирование*.

Доклад В.М.Глушкова состоялся в актовом зале старого корпуса Академии. Людей было очень много, стояли даже между рядами кресел. Во время доклада хозяева помещения пытались увести людей с середины зала, предупреждая, что здание старое и пол может провалиться. Но доклад продолжался и никто не покинул помещения.

Секционные доклады были представлены в 13 секциях: Алгебра, Теория чисел, Геометрия, Топология, Теория функций, Функциональний анализ, Теория вероятностей и математическая статистика, Обыкновенные дифференциальные уравнения, Уравнения в частичных производных, Математическая физика, Математическая логика и основания математики, Вычислительная математика, История математики.

3.3. "Программа 4-ого Всесоюзного математического съезда" в Киеве

Сотрудники ВЦ выступали в секции "Вычислительная математика". Приводится названия докладов руководителей разных лет и уровней. (В.М. Глушкову и С.Л. Соболеву – 50мин, остальным – по 15 мин.)

В.М. Глушков, С.Л.Соболев. *Некоторые математические проблемы теории обучающихся автоматов*

А.А. Стогний, Н.М. Грищенко. *Обучающаяся алгоритмическая система для распознавания осмысленных предложений*

Л.Н. Иваненко. *О применении машин для конформного отображения односвязных однолистных областей*

А.А. Летичевский. *Условия полноты системы конечных автоматов*

А.А. Стогний. *Некоторые математические вопросы построения цифровой логической машины*

Е.Л. Ющенко-Рвачева. *Адресный язык и проблема автоматизации программирования*

В.С. Михалевич, И.З. Шор. *Метод последовательного анализа вариантов решения вариационных задач управления, планирования и проектирования*.

А.А. Ляпунов. *Обзор проблематики кибернетики.*

С.В. Яблонский. *Обзор математических проблем кибернетики.*

О.С. Кулагина. *Об экспериментах по машинному переводу и о машинной выработке переводческих алгоритмов.*

А.П. Ершов, С.С. Лавров. *Об экономии памяти при составлении программ.*

Э.З. Любимский. *Расшифровка выражений типа АЛГОЛ* .

Академик С.Л. Соболев. *Расшифровка письменности Майя.*

Р.Х. Зарипов. *Об алгоритмизации процесса сочинения музыки.*

Позже Зарипов Р.Х. приезжал в ВЦ с докладом о создании мелодий с помощью ЭВМ. Между прочим, расказывал о том, как известный автор очень популярных песен 1960-70-х годов создает новые мелодии. Оказалось, очень простым способом: с той или иной имеющейся песни регулярным образом вычеркивает ноты, например, каждую пятую. Потом, проигривая то, что осталось, получает новую мелодию, которую оценивает с точки зрения, можно ли из нее сотворить песню. Р.Х. Зарипов даже демонстрировал одну знакомую песню на акордеоне, полученную таким способом из другой, которую тоже играл. Если это правда, то как он об этом узнал, неизвестно: может, сам подсказал композитору такой нехитрый способ. Так или иначе, но его доклад имел успех у слушателей.

В сентябре - октябре 1961 года В.М. Глушков поручил Зарипову В.Х. на семинаре в отделе выступить с рефератом статьи о так называемом универсальном решателе проблем, по-английски General Problem Solver (GPS). В то время в записную книжку автора было занесено несколько ссылок на зарубежные статьи по искусственному интеллекту, в том числе о GPS и игре с компьютером в шахматы. Вероятно, реферировалась одна из следующих двух:

A.Newell, J.C. Show and H.A.Simon. Report on a general problem solver. Proc. of the Intern. Conf. on Information Processing. 1959. pp. 256-264.

A.Newell, J.C. Show and H.A.Simon. A Variety of Intelligence Learning in a General Problem Solver, Proc. of Meeting on Self Organizing Systems. Pergamon Press, 1960.

К тому времени автор лишь начала изучать английский язык (под влиянием заказа на реферат), но справилась со статьей благодаря помощи по английскому языку Л.А. Калужнина. Основная изложенная в статье идея демонстрировалася примером применения соотношений типа <левая часть = правая часть> к алгебраическим выражениям с целью их упрощения. В этом случае в качестве соотношений использовались формулы для квадрата суммы, разности и др. Итак, в выражении ищем подвыражение, совпадающее с левой частью соотношения, начиная с первого. Не найдя, применяем следующее соотношение, а найдя, заменяем найденное подвыражение на правую часть текущего соотношения. В этом случае следующий шаг: к измененному алгебраическому выражению применяем соотношения, снова начиная с первого. И так до тех пор, пока ни одно их соотношений неприменимо.

Интересно, что рассказывая содержание статьи у доски, докладчица удивлялась, когда В.М., опережая ее, делал выводы из услышанного, совпадавшие с выводами в статье, которые она должна была огласить в следующую минуту. В одном месте статьи, где ей предстояло пожаловаться на то, что она не поняла выводов из только что сказнного, В.М., опередив ее жалобу, рассказал о том, что было ей непонятно. А одна ее оплошность развеселила слушателей и В.М. – не "узнала" фамилии французского математика Коши, написанной на французском языке.

Однажды осенью 1961 года на своем рабочем столе автор этого текста среди бумаг случайно обнаружила записку Виктора Михайловича с пометкой рукой А.А. Стогния "*Наде*". Вот ее текст:

Тривиальный алгоритм обучения распознаванию осмысленных фраз. Алгоритм с экстраполяцией опыта. Выработка понятий. Корелляционные связи между понятиями и эффективность обучения. Поиск эффективных алгоритмов обучения на основе статистических испытаний.

Эксперименты проводились на ЭВМ "Киев" по заданию А.А. Стогния. Н.М.Грищенко приобрела опыт обучения компьютера распознавать осмысленные предложения естественного языка. Воспоминание об этом приведено в книге [9, с.71].

.

3.4. Работы по распознованию текстов на М-20, ПК «Мир» …

Работы по распознаванию смысла фраз на русском языке, т.е. в области семантических сетей , как теперь это называется. Этим занимался А.А. Стогний и частично А.А.Летичевский, который сделал алгоритм, , а А.А. Стогний подготовил хорошие программы. (они хранились на бланках машины "Киев"). По потоку предложений на входе этот алгоритм строил семантическую сеть, т. е. определял, какие слова с какими корреспондируются. Например, "Стул стоит на потолке" хоть и правильно грамматически, но семантически неверно и т. д. Были сделаны зачатки картины мира, причем было придумано экономное кодирование, затем А.А. Стогний переключился на распознавание дискретных образов, тематику Ю.И. Журавлева, да и я оставил это дело, и у нас оно захирело. Надо было его с машинным переводом связать, но опять не хватило людей, а я не могла заниматься семантической алгоритмикой. Летичевский А.А. сделал в 1962 году доклад в Мюнхене на конгрессе IFIP на эту тему, это стало сенсацией – у американцев ничего подобного не было. Тогда же его избрали в программный комитет Международной федерации по обработке информации. (О докладе в Мюнхене см. ниже)

В 1962 году решалась задача моделирования этой задачи на машине "МИР", ЭВМ "Киев" иа М-20.

Под руководством В.М. Глушкова в 1961 году сотрудниками его отдела Э.Ф. Скороходько и Л.Э. Пшеничной была выполнена работа по синтезу осмысленных предложений на ЭВМ, результат которой представлен в статье "*Синтез осмысленных предложений на ЭЦВМ*", напечатанной в сборнике [10]. В статье есть ссылка на алгоритм В.М. Глушкова распознавания осмысленности предложений, как на инструмент для проверки правильности результатов синтеза осмысленных предложений.

Все так называемые логические программы для ЭВМ того времени разрабатывались с немалыми трудностями, связанными с ориентацией ЭВМ только на вычисления и с отсутствием устройств ввода-вывода символов, отличных от чисел и знаков арифметических операций. Поэтому возникла необходимость в усовершенствовании компьютеров в соответствии с требованиями, которые возникают по мере расширения нечисловых областей применения компьютерной техники. Перечень таких требований и их обоснование были основной темой кандидатской диссертации А.А. Стогния.

В то время кибернетика еще не была признана математиками старшего поколения самостоятельным разделом в математике. Они считали, что вычислительная техника – это лишь вспомогательная ветвь математики. Защита диссертаций по кибернетике вызывала сопротивление в тогдашних математических кругах математиков. Известно, что А.П. Ершов защитил кандидатскую диссертацию почти через два года после ее написания. У него даже были проблемы с оппонентами. Такая же судьба могла постичь и А.А. Стогния. Поэтому в январе 1962 года научный руководитель А.А. Стогния В.М. Глушков обратился к Ю.Г. Косареву, ученому из Института математики СО АН СССР с просьбой поддержать А.А. Стогния, прислав отзыв на его диссертацию. Об этом свидетельствует письмо В.М. к Ю.Г. Косареву от 12.12.1961 год (Архив академика А.П. Ершова [11]), в котором он обосновывает свою просьбу:

...В результате Стогний А.А. подготовил кандидатскую диссертацию на тему о рациональных принципах построения универсальных цифровых машин для преобразования буквенной информации. Смысл ее состоит в том, что на основе исследования программ для нескольких изученных им логических алгоритмов, он дает ряд рекомендаций для построения памяти машин и набора операторов, делающих эти машины значительно более пригодными для решения неарифметических задач, чем все существующие ныне машины.

Защищался защищена в феврале этого года в Киеве, на объединенном ученом совете отделения физ.-мат. наук. Официальные оппоненты А.А. Ляпунов и Л.А. Калужнин дали хорошие предварительные отзывы. Получен также отзыв от ВЦ АН СССР.

После этого наметилась значительная оппозиция из числа абстрактных математиков, входящих в наш совет, которая считала всю математику, связанную с программированием и машинами не математической, и чуть ли вообще не научной. Для того чтобы разбить это мнение (с которым Вы должны быть знакомы и по сибирскому опыту), нам необходимо прежде всего заручиться поддержкой крупных математических организаций, развивающих прикладную тематику, к числу которых относится прежде всего Института. В результате был получен отзыв на диссертацию А.А. Стогния от имени Института математики СО АН СССР. Отзыв, подготовленный А.П. Ершовым, был получен, и В.М. Глушков поблагодарил А.П. в письме от 11.02.1962 года, а также поздравил его самого с успешной защитой кандидатской диссертации.

В феврале 1962 года успешно прошла защита кандидатской диссертации А.А. Стогния на тему "*Исследование рациональных принципов построения универсальных цифровых машин для преобразования буквенной информации*".

Из автореферата: "*В первой главе дается сжатое описание программ, составленных автором для решения на ЦВМ некоторых логических задач*", среди которых наболее полно охарактеризован алгоритм и эксперименты по обучению машины распознавать осмысленность простых предложений естественного языка.

Диссертация А.О. Стогния была актуальной, содержала детально отработанные и обоснованные рекомендации по архитектуре логической машины. Об этом свидетельствует подготовленная за материалами диссертации его статья "*Один вариант структуры цифровой машины для преобразования буквенной информации*", напечатанная в сборнике [10].

Однако, через сложившиеся тогда обстоятельства логическая машина не была построена. Через несколько лет структура компьютеров была усовершенствована за счет устройств ввода-вывода символьной информации, организации байтовой структуры памяти, а также появились

языки программирования высокого уровня со средствами работы с символьной информацией, что значительно облегчило применение компьютеров для решения так называемых логических задач.

К середине 1962 года появились результаты исследований В.М. Глушкова и в других областях искусственного интеллекта, о чем свидетельствуют публикации [12], [13], [14], а также доклад на конгрессе в Мюнхене.

С 27 августа до 1 сентября 1962 года состоялся конгресс IFIP-62 (Мюнхен), в рамках которого был проведен симпозиум по искусственному интеллекту, где выступил В.М. Глушков с докладом "*Некоторые вопросы теории самообучения машин*" [15]. Во время виступления среди нескольких тем, которые развивались в ВЦ, главное внимание В.М.Глушков сосредоточился на эксперименте по самообучению компьютера распознавать осмысленные предложения естественного языка. Тема его доклада подается ниже в дословном переводе с английского языка, а другие темы только называются.

Во время моделирования мыслительных процесов на компьютерах становится очень важным вопрос самообучения машин... В Институте кибернетики АН УССР в Киеве проводится работа по самообучению машин с учетом различных аспектов теории.

Первый аспект касается общих вопросов теории самообучения...

Второй аспект работы касается обучения автоматов распознаванию образов...

Третий аспект, наиболее интересный с точки зрения моделирования мыслительных процессов, касается построения систем для обучения распознаванию смысла предложений. Конкретно, рассматривается русский текст сравнительно простой грамматической структуры "субъект – глагол – предлог – объект". Для экспериментов выбирали сравнительно небольшой словарь (приблизительно 100 слов). Это дало возможность использовать 1000 ячеек памяти, включая место для программы и различных таблиц.

Эксперимент был организован следующим образом. Сначала, в дополнение к программе, ввели словарь из 100 русских слов. Они были выбраны случайно, таким образом были получены 70 существительных, 20-25 глаголов и 5-10 предлогов. Из выбраных слов были построены случайным образом предложения при выполнении двух условий: они должны быть осмыслены и иметь конструкцию, не сложнее указанной выше. После генерации достаточного числа предложений мы последовательно ввели их в компьютер. После обработки этой информации компьютер сам сконструировал несколько предложений и спросил программиста, осмыслены ли они.Требовалось, чтобы программист дал правильные ответы на эти вопросы. После этого машина переключалась на режим экзамена. С того же словаря экзаменатор построил новые предложения (осмысленные и неосмысленные) и попросил компьютера ответить, какие из них осмысленны. В реальном эксперименте материал, на котором было произведено обучение (включая предложения, построенные самой машиной), составлял около трети всех возможных осмысленных предложений. В большинстве случаев компьютер правильно классифицировал остальные две трети предложений.

Основную идею легче понять на простом примере предложений, имеющих структуру "субъект – глагол". Сначала компьютер просто накапливает вводимые осмысленные предложения, но специальный параметр, встроенный в программу и называемый коэффициентом сдерживания (coefficient of retentivity), рано или поздно изменяет содержимое памяти. Если, например, этот коэффициент равен 2, компьютер может принять не больше двух предложений с одним и тем же глаголом, например, "профессор думает" и "студент думает". Если позже поступит новое предложение с тем же глаголом, например, "мальчик думает", то машина определяет новое выражение (название) для класса "думающих", и записывает, что "профессор", "студент" и "мальчик" – члены класса "думающих".

Второй специальный параметр программы – коэффициент осторожности (coefficient of caution) – регулирует процесс переноса свойств от одного класса к другому. Например, если этот коэффициент равен 2, тогда достаточно компьютеру распознать, что два члена класса

"думающих" могут говорить, чтобы заключить, что все думающие могут также и говорить. Естественно, что при экстраполяции, основанной на неполной индукции, могут возникнуть ошибки. Однако, следующий блок программы генерирует случайным образом осмысленные (с точки зрения машины) предложения и затем автоматически исправляет ошибки, формируя новые классы существительных, основанные на одном, двух, трех или любом числе свойств.

С помощью экспериментов можно выбрать наилучшие значения коэффициентов, при которыхпроцесс обучения машины происходит за кратчайшее время. Изменяя значения этих параметров, можно получить разные схемы поведения процесса обучения от простого механического запоминания (зубрежки) до тенденции к неконтролируемой фантазии.

Эксперименты, моделирующие на компьютере некоторые из простейших процессов биологической эволюции, принадлежат к четвертому типу исследований, проводимых в Киеве...

Вскоре после конгресса В.М. Глушкову пришло письмо от американского ученого Saul Amarel с просьбой прислать материал об этой работе. Приводим письмо на языке оригинала с переводом на русский язык:

Radio Corporation of America RCA Laboratories
Professor V.M. Glushkow
Institute of Cybernetics Academy of Science
Lysogorskaya, 2 Kiev, USSR

During your interesting presentation at the Artificial Intelligence Symposium of the Munich IFIP-62 Congress, you mentioned that papers covering your research have been presented at the 1961 Kiev Symposium on Self Organizing Systems.

Симпозиуме по искусственному интеллекту в Мюнхене конгресса IFIP-62 выступил В.М.Глушков по самоорганизующимся системам в рамках исследований по искусственному интелекту в ИК АН УССР:

As I am actively interested in artificial intelligence research I shall greatly appreciate receiving reprints of your papers at the Kiev Symposium, as well as other material that you may have published on your recent work. I am especially interested in mechanisms for concept formulation and I would like to know more about your project on machine recognition of meaning of phrases that are assembled from a limited and controlled language.

Поскольку я очень заинтересован в исследованиях по искусственному интеллекту, то был бы очень благодарен вам за получение репринта вашего доклада на Киевском симпозиуме, а также других материалов о ваших последних работах. Особенно интересуюсь механизмами формирования понятий и хотел бы также узнать больше о вашем проекте машинного распознавания осмысленности фраз, составленных из ограниченного языка.

Sincerely
Saul Amarel
Head of Computer Theory Group in the RCA Laboratories (in Princeton)

Справка:

Saul Amarel (1928–2002) был профессором университета Рутгерс, он наиболее известен благодаря своим пионерским работам по искусственному интеллекту (ИИ). Взнос Saul Amarel в компьютерные науки и ИИ происходил на протяжении всего развития этой области, начиная с 1950-х годов..

Позже Saul Amarel объединил ИИ с другими областями. Он, в частности, разрабатывал машину, которая могла диагностировать болезни.

Amarel был награжден премией Алена Ньюэла от ACM за значительный вклад в исследования ИИ, особенно в продвижении нашего понимания роли изображений в решении проблем и в теории, и в практике планирования вычислений. Он стал членом IEEE.

В монографии [6, с.520] В.М. Глушкова, искусственному интеллекту посвящена отдельная глава, в которой В.М. связывает проблему определения семантики текстов на естественных языках с помощью компьютеров с проблемой осмысленного диалога человека с машиной. Цитируем:

...Проблема понимания текстов на естественных языках не может считаться до конца решенной, если предназначенная для этих целей автоматическая система не способна вести осмысленный диалог с человеком и, самое главное, обучаться в результате диалога.

При ведении такого диалога важно уметь осуществлять автоматический перевод с внешнего языкового представления на язык семантической сети и обратно. С этой целью удобно использовать аппарат формальных семантических грамматик с процедурами классификации и объединения языковых оборотов, равнозначных по смыслу.

....Следует заметить, что построение семантических грамматик в значительной мере облегчается применением процедур, строящих семантическую классификацию в результате анализа предъявляемых системе правильных и неправильных фраз (как с точки зрения синтаксиса, так и с точки зрения семантики). Один из способов такой классификации был предложен автором еще в 1961 году...

Начав широким фронтом развивать идеи создания систем искусственного интеллекта на машинах первых поколений, В.М.Глушков надеялся достичь видимых результатов на протяжении ближайшего десятилетия или двух. Вопреки ожиданиям, В.М. достаточно быстро пришел к заключению о том, что сложные кибернетические системы требуют многолетнего труда. Чтобы правильно организовать эту работу, В.М. выдвинул принцип "единства ближних и дальних целей", касающийся времени, необходимого для создания сложных кибернетических систем.

Лучше всего содержание этого принципа изложено в его воспоминаниях, которые он продиктовал в последние дни своей героической жизни. Приводим отрывок воспоминаний из книги [9, с.45]:

«Дело заключается в том, что в кибернетике есть одна особенность. Когда развивались другие науки, которые не имели дела со столь большими системами, как кибернетика, то обычно возникновение идеи о том, как решить задачу (особенно в математике), являлось главным. Это было 90% дела. Если идея была верной, то ее оформление занимало 10%. В биологических исследаваниях эти цифры могут быть другими: 40% – идея, а 60% – труд на ее реализацию. А в кибернетике получается так, что в некоторых случаях идея составляет около 0,01%, а все остальное – 99,9% – это ее реализация. Объясню это на примере. Мы с самого начала стали развивать направление, называемое искусственным интеллектом, связанное с построением разумных машин и соответствующих программ. На эту тему я написал книгу "Теория самоусовершенствующихся систем", и во "Введении в кибернетику" ряд разделов был посвящен специально этому вопросу.

Такая *недооценка сложности кибернетических задач типична для периода становления любой науки. Я как-то быстро (может потому, что занимался философией в свое время) это понял и таких ошибок не делал, таких предсказаний не давал. Особенность больших систем в том, что от идей по их построению до их реализации очень длительный путь. Отсюда и появился важный управленческий принцип – единства дальних и ближних целей.*

Данный принцип формулируется так: в новой науке, в кибернетике, не следует заниматься какой-то конкретной ближней задачей, не видя дальних перспектив ее развития. И наоборот, никогда не следует предпринимать дальнюю перспективную разработку, не продумав ее разбить на этапы, чтобы каждый отдельный этап, с одной стороны, был шагом в направлении к этой большой цели, а вместе с тем он сам по себе смотрелся как самостоятельный результат и приносил конкретную пользу».

Начатые В.М. Глушковым работы по распознаванию машиной осмысленности предложений на естественном языке продолжены в других вариантах, поскольку этого требуют потребности информатизации общества.

Остановимся кратко на исследованиях в этом направлении, начатых в 1960-х годах в отделе В.М. Глушкова его сотрудниками-лингвистами Э.Ф. Скороходько и Л.Э. Пшеничной. В.М. заинтересовался результатами их работы, пригласив выступить на семинаре в отделе. Л.Э. Пшеничная вспоминает, с каким интересом слушал В.М. ее доклад о результатах исследований лексической семантики естественного языка, которые были опубликованы в статьях.

В. Ф. Скороходько опубликованы в монографии [16], приводя исследования семантических сетей и построения лексикона и текстов. Во введении написано:

Данная работа посвящена одному из наиболее важных и перспективных направлений в современной лингвистической семантике – сетевому моделированию языка. Подобная тема впервые является объектом монографического исследования в советской науке. Этот метод исследования основывается на построении семантических сетей, моделирующих смысловую сторону лексики и текста. ...Семантическая сеть оказалась особенно полезной при решении многих теоретических и практических задач, особенно связанных с проектированием лингвистического обеспечения систем искусственного интеллекта.

Проблема построения семантических сетей лексики и текстов состоит в том, что семантическая связь между словами, а тем более между фрагментами текста, невидима – в отличие от видимых морфологического строения слов и синтаксической связи между словами в предложении. Поэтому задача лингвистов состоит в том, чтобы найти семантическую связь между словами и выразить ее в удобной для использования форме.

Предметом рассмотрения в монографии Э.Ф. Скороходько – **семантические связи между словами терминологической лексики и между предложениями научных текстов**. Подаем краткую общую схему исследований без точных формулировок понятий разработанной автором монографии лингвистической теории. Автор утверждает, что для полного описания семантической связи между двумя словами необходимо указать *содержание* связи, ее *направление* и *силу*. Содержание отображает соотношение, которое объективно существует между смыслами в виде категорий типа **"субъект – действие", " часть – целое", "причина – следствие"** и пр. Порядок следования категорий свидетельствует о направлении связи, например, "причина – следствие" в противовес "следствие – причина". Сила связи характеризирует расстояние между словами, рассматриваемое в конкретном семантическом пространстве.

Обнаружить семантические связи между словами можно методами двух типов: формальными, основанными на корреализационной зависимости между семантической связью и размещением слов в тексте, и неформальными – с использованием интуиции носителя языка.

Формальные методы имеют то преимущество, что они свободны от субъективизма; однако их недостаток состоит в том, что с их помощью можно определить лишь наличие и силу семантической связи.

Избавиться от вышеупомянутого недостатка можно, приняв во внимание, что каждое полнозначное слово содержит одну или несколько семантических составляющих. В качестве примера семантических состаляющих в монографии рассматривается описание слова "*туман"* в словаре Ожегова: "*непрозрачный воздух, насыщенный водяными парами, а также загрязненный пылью, дымом, копотью*". В описании слова *туман* присутствуют такие семантические состаляющие: 1) *непрозрачный воздух*; 2) *насыщенный водяными парами*; 3) *загрязненный пылью, дымом, копотью*. Множество семантических составляющих слова образуют его лексическое значение (смысл). Между парой слов существует семантическая связь, если их семантические значения имеют общие семантичские состаляющие.

Чтобы определить семантическое значение слов некоторой терминосистемы с целью построения ее семантической сети, в монографии предложен метод, в котором используется

неформальная семантическая информация, наболее полно представленная в терминологических толковых словарях соответствующей области знаний. Однако следует иметь в виду, что таким словарям в большей или меньшей мере присущ субъективизм и наличие ошибок в определениях терминов. Цитируем автора монографии Э.Ф. Скороходько:

Толковый словарь как способ представления системы семантических связей и семантической структуры лексики... не обеспечивает однозначного и эксплицитного представления семантических связей, не позволяет прослеживать глубинные связи, т.е. цепочки связей (в этом одна из коренных причин многих логических ошибок в словарях, на которые не раз указывалось), мало пригоден для определения количественных характеристик и т.д.

Идеальной математической моделью любого системно-структурного образования является граф... Весьма удобен граф и для изображения системы семантических связей и семантической структуры лексики.

Вершины графа – значения (смыслы) слов, а ребра – семантические связи между значениями слов. Содержание семантических связей можно передавать на ребрах словами, числами или буквами. Направление связи совпадает с направлением ребра, а силу связей можно обозначать на ребрах условными знаками. Отметим, что начальной информацией для построения графа семантических связей между терминами являются термины из толковых словарей с выделенными семантическими составляющими.

Введение в компьютер лексической семантики терминосистемы является, по существу, обучением машины, хотя и не в прямом диалоге "человек – машина", что было бы весьма неэффективным, учитывая объем информации, котрую нужно ввести. Диалог становится актуальным, когда идет проверка правильности сети на примерах осмысленных и неосмысленных фраз, подаваемых машине для распознавания их осмысленности с помощью семантической сети.

Терминологические семантические сети проверяют правильность определения терминов в терминологических словарях, оценку словарей по словарным параметрам. Семантические сети лексики являются базисом для построения семантических сетей, которые используются в процессах автоматизации индексирования и реферирования текстов. Результат этих процессов, в свою очередь, является основой для формирования поисковых образов документов – запросов к системам поиска текстовых документов в Интернете.

Актуальность исследований в этом направлении оценивается в период бурного развития различных типов сетей (Интернет, банковские сети, макроконвейерные и др.). В каждой их них диалог компьютера с человеком является одной из главных частей их математического обеспечения, их сутью, благодаря чему обычный человек может пользоваться компьютером и услугами сетей без специальной подготовки, не тратя времени на изучение математического обеспечения и вычислительной техники. Но путь к этой простоте был очень нелегким. Отметим, что украинские математики во главе с В.М. Глушковым были одними из первых, кто увидел перспективу таких исследований, и начали разрабатывать фундамент интеллектуальной среды..

Таким образом, По управлению Глушкова по ИИ блии защищены диссертации Стогний А.А., показаны работы по искусственному интеллекту сотрудниками отдела

Список литературы

1. В.М. Глушков "Теория алгоритмов". Изд-во КВИРТУ, 1961, 167с.

2. E.M. Zaitzeff and M.M. Astrahan. Russian Visit to U.S. Computers in IRE Transactions on electronic computers, Vol EC-8, Dec., 1959, Numb.4, pp. 489-496.

3. Тезисы докладов 2-ой научной конференции по вычислительной математике и вычислительной технике (6-10 июня). Издание ВЦ АН УССР. Киев – 1960. 60 стр.

4. IRE Transactions on Electronic Computers", Oct., 1961. (pp. 759-776)

5. Глушков В.М. Введение в теорию самосовершенствующихся систем.- Киев : Изд-во КВИРТУ. – 1962, 109 с.,

6. Глушков В.М.Основы безбумажной информатики. – Москва: "Наука". – 1982, 562 с.

7. В.М. Глушков, Е.Л. Ющенко "Вычислительная машина "Киев". Математическое описание" Гос. издат. техн. лит.УССР. Киев, 1962, 182 с.

8. "Принципы построения самообучающихся систем", Гос. издат. техн. лит. УССР, Киев-1962.- 118с.

9. Б.Н. Малиновский "Академик В. Глушков", Київ, Наук. думка, 1993, 142 с.

10. Сб. "Проблемы кибернетики", вып. 10, М., Физматгиз, 1963.

11. http://www.ershov.ras.ru/russian/arch.html

12. Универсальная установка для исследования алгоритмов распознавания изображений // Принципы построения самообучающихся . систем. Киев.-1962. С. 63-72.- Глушков В.М., Ковалевский В.А., Рыбак В.И.

13. Алгоритм обучения машины распознаванию простейших геометрических фигур // Там же.- С. 5-18. - Ковалевский В.А., Рыбак В.И.

14. К вопросу о самообучении в персептроне // Ж. вычисл. мат. и мат. физ.-1962, № 6.- С. 1102-1110.

15. Certain Questions of the Theory of Machine Self-learning // Proc. IFIP Congress- Munich.1962, P. 480-481

16. Э.Ф. Скороходько. Семантические сети и автоматическая обработка текста. – Киев, Наук. думка, 1983. – 213

Развитие идей ИИ и технологии программирования В.М. Глушкова в отделе Глушкова с 1955 года (укр.).

Мищенко Н.М.

.

Періоди: Феофанівський (1956-1958), Лисогірський (1959-1964), Великокитаївський (1965-1975), Теремківський - 1 (1956-1990)

Коротка характеристика діяльності по программированию в цей період

В С Т У П

Феофанівський період: технік-обчислювач, працювала на настільних електричних калькуляторах, інколи виконувала функції оператора на МЭСМ.
Лисогірський період: програміст**,** посади: інженер, згодом молодший науковий співробітник. Виконано програмування чотирьох різногалузевих завдань нечислового характеру на ЕОМ "Киев". Перша програма – за алгоритмом морфологічного аналізу І.О. Мельчука, молодшого наукового співробітника Інституту мовознавства АН СРСР, друга – Програмуюча програма -2 (ПП-2) для ЕОМ "Киев" за алгоритмом старшого інженера Л.М. Іваненка та зав. відділом програмування К.Л. Ющенко, третя – навчання машини "Киев" розпізнавати осмисленість речень натуральною (російською) мовою за алгоритмом В.М. Глушкова:, четверта – моделювання машини "МИР" на машині "Киев" запрограмовано за власним алгоритмом.
Великокитаївський період: розробляння розширних мов програмування алгоритмів проектування електронних пристроїв та побудова трансляторів з цих мов для ЕОМ М-20 та М-220 з колективом програмістів відділу Теорії цифрових автоматів В.М Глушкова. Захист кандидатської дисертації за результатами досліджень та програмування.

Теремківський період -1 (1976-1990): автоматизація програмування для формальних мов, що описуються модифікованими формами Бекуса-Наура (МБНФ). Інструментальна саморозширна система програмування ТЕРЕМ, за допомогою якої у відділі ТЦА з 1976 року, а з 1980 року – у відділі РВМ (рос.мовою – рекурсивных вычислительных машин) розроблялося кілька систем програмування для Макроконвейєра – обчислювального комплексу на базі машин ЄС ІВМ.

Теремки. Вхід до Інституту кібернетики ім. В.М. Глушкова НАНУ (1982 рік).

Институт Кибернетики.

11 травня 1976 року настав новий період моєї діяльності в ІК АН УРСР: нове місце роботи –

Теремки, нові машини ЄС, побудовані в СРСР за зразками американської фірми IBM, нова тематика в Інституті, нова квартира – теж у Теремках у 10 хвилинах бігу до роботи.

У відділі Теорії цифрових автоматів (ТЦА), зав. відділом академік В.М. Глушков, розпочата робота по програмуванню системи ПРОЕКТ-ЄС на новій ЕОМ ЄС 1060. Паралельно виконувалися роботи з формування мови програмування та спеціальної операційної системи для нового високопродуктивного многопроцесорного обчислювального комплексу (абревіатураосійською мовою – МВК) з макроконвейєрною організацією обчислень, технічний проект якого

завершувався. У цьому проекті машині ЄС 1060 відводилася роль периферійного процесора. Тож його математичне забезпечення (рос. – МО) мало бути складовою МО МВК.

Для автоматизації побудови математичного забезпечення системи ПРОЕКТ-ЄС та МВК були створені інструментальні засоби у вигляді розширних систем програмування (РСП), відповідно,

Т-ЄС та ТЕРЕМ .

І. Інструментальні засоби побудови систем програмування

Насамперед розглянемо типову схему трансляції програм мовою програмування, що містить засоби її розширення. На Схемі 1 представлена РСП, вхідна програма для якої формується з використанням трьох мов, притаманних для кожної РСП, а саме: мова L_B – базисна мова програмування, L_e – мова для введення об'єктів-розширень базисної мови, L_d – введені об'єкти-розширення, використані у програмі.

Прообразом системи, поданої на Схемі 1, є РСП Т на М-220 для системи ПРОЕКТ.

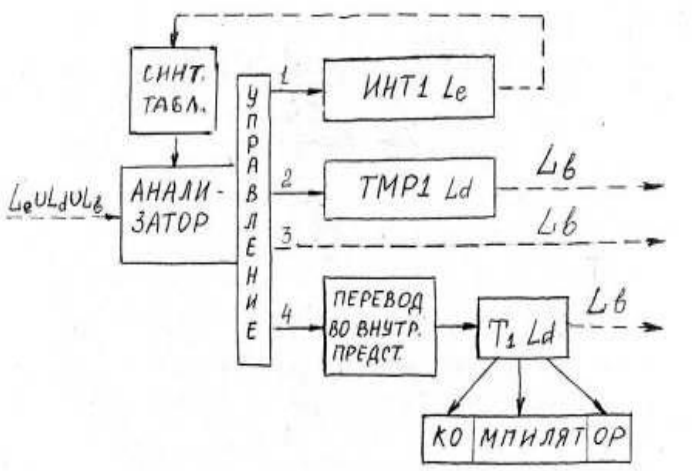


Схема 1. Загальна схема розширної системи програмування

Фрагменти програми, де метамовою L_e вводиться опис розширень, повинні передувати використання введених нових об'єктів у цій же програмі.

У прямокутниках на Схемі 1 – позначки процесорів обробляння програм, написаних розширеною мовою. Розглянемо 4 варіанти дій РСП.

(1) ИНТ1 – программа розширення синтаксичних таблиць за описом нових операторів мовою L_e;

(2) TMP1 – макропроцесор, який обробляє оператори розширення L_d;

(3) – запис об'єктів базисної мови в результат без змін;

(4) T_1 – компілятор, що містить засоби трансляції тих об'єктів розширення L_d, для перекладу яких на базову мову засобів макропроцесора не вистачає. Цей компілятор містить засоби систем побудови трансляторів (СПТ).

Конфігурації складових (1) – (4) можуть бути різними. Схему без складової 4 зазвичай використовують у препроцесорах. В кількох трансляторах для МВК присутня лише складова (4), тоді як препроцесор "Расширитель" до цих трансляторів містить лише складові (1) – (3).

РСП Т-ЄС була реалізована мовою ПЛ/1 у **1978-1979** роках за схемою РСП Т для М-220, реалізованої протягом Великокитаївського періоду, але без складової (4). РСП Т-ЄС використовувалася для побудови процедурних розширень мови ПЛ/1 ЄС ЕОМ, призначених для програмування системи ПРОЕКТ-ЄС. Вона характеризувалася такими властивостями:

1. Засіб розширення: техніка синтаксичних макросів. У загальному випадку розрізнялися дві метамови Ld і Lt для опису, відповідно, синтаксису і семантики розширень. Вони є підмножинами вхідної мови РСП у її початковому стані. В РСП Т-ЄС базисна мова, мова програмування системи РСП Т-ЄС і мова Lt опису семантики розширень одна і та ж – ПЛ/1.

2. Ефективність і зручність роботи користувача з системою, що дозволяє здійснювати розширення порівняно легко – шляхом додавання нових об’єктів до базисної мови.

3. Гнучкість системи, яка забезпечується табличним зв’язком між об’єктами синтаксичного та семантичного описами мови. Програмне управління цим зв’язком дозволяє оперативно вносити зміни до описів синтаксису та семантики.

4. Багаторівневість процесу розширення. Основний принцип при побудові чергового рівня вхідної мови – максимальне використання вже побудованих засобів.

, в РСП Т-ЄС засобами макротехніки можна вводити у вхідну мову об’єкти, що виражаються в термінах базисної мови та вже побудованих розширень і не потребують зміни її транслятора. Зауважимо, що в РСП Т-ЄС можна використовувати базисні мови, які допускають найпростіші методи аналізу текстів, а саме, розпізнавання об'єктів базисної мови за ключовими словами, якими починаються її оператори та описи змінних. За таким же принципом у 1989 році була побудована РСП Т-МВК, як препроцесор до системи програмування МАЯК для МВК.

У наступному розділі подається опис реалізації та приклад розширення мови ПЛ/1 операторами управління даними СОСУД системи ПРОЕКТ-ЄС.

РСП ТЕРЕМ, побудована у 1980-1982 роках – система побудови трансляторів (СПТ) для реалізації мов програмування, синтаксис яких описується формами Бекуса-Наура.

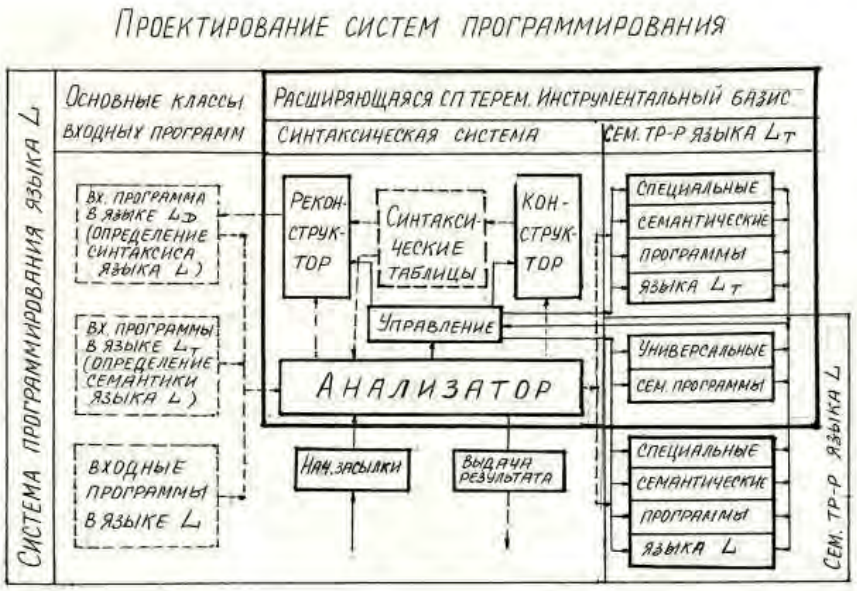


Схема 2. Використання РСП ТЕРЕМ для реалізації мови програмування з умовною назвою L

На Схемі 2 представлена РСП ТЕРЕМ. Ядро системи у вигляді великого прямокутника грубою лінією зверху праворуч містить:
1) синтаксичну підсистему – Аналізатор, який виконує лівосторонній низхідний аналіз з обмеженими поверненнями і побудову дерева аналізу;
2) спеціальні семантичні програми мови $L_т$, що виконують розширення вхідної мови засобами СПТ;
3) універсальні семантичні програми, в яких реалізовані найуживаніші функції перекладу;
4) процедуру обходу дерева аналізу, яка керує виконанням семантичних процедур.

Квадрат грубою лінією у нижньому правому куті схеми показує програми, які мусить розробляти замовник у випадку, коли необхідні спеціальні засоби трансляції.
Основний принцип застосування системи для реалізації деякої мови **L**: використання РСП ТЕРЕМ у ролі **ядра нового транслятора і програм для розширення цього ядра** об'єктами двох типів: а) синтаксичними таблицями мови **L**, побудованими самою РСП за описом її синтаксису мовою модифікованих Бекусово-Наурівських форм; б) якщо необхідно, спеціальними семантичними процедурами, побудованими користувачем, для яких система пропонує необхідне інформаційне оточення і керує їх виконанням.. Результат: готовий мовний процесор - компілятор.

Синтаксично-керовані переклади реалізуються стандартними семантичними програмами. Семантичні програми для реалізації атрибутних перекладів розробляються користувачами
За допомогою РСП ТЕРЕМ були реалізовані мови програмування сімейства МАЯК для многопроцесорного обчислювального комплексу (МВК) з макроконвейєрною організацією обчислень. Для автоматичного внесення змін до вхідних програм трансляторів з мов сімейства МАЯК у 1989 році був побудований препроцесор до них на базі РСП Т-ЄС, розширеної модулем об'ємом 344 оператори мови ПЛ/1.

Практика реалізації кількох мов сімейства МАЯК показала, що використовуючи РСП ТЕРЕМ для їх реалізації, доводилося програмувати лише 3/8 об'єму необхідного транслятора, отже, 5/8 припадає на універсальні модулі РСП ТЕРЕМ.
РСП ТЕРЕМ можна використовувати також для реалізації підмножин природних мов у людино-машинних системах, які еволюціонують в процесі реалізації та використання.

Підсумуємо суттєві властивості РСП ТЕРЕМ.

1. Склад системи ТЕРЕМ формується за рахунок програмних модулів, у яких реалізовані

універсальні функції мовних процесорів (***накопичення знань***).

2. Принцип застосування системи: збірка ядра майбутнього мовного процесора з готових модулів (***зберкове програмування***) і, якщо необхідно, розширення ядра модулями, створеними спеціально для мови, що реалізується.

3. Система забезпечує всі модулі спільним інформаційним полем і засобами доступу до нього (***спільна платформа модулів***), а також засобами управління взаємодією модулів, які тиражуються у кожний мовний процесор, побудований за допомогою системи ТЕРЕМ.

Виділені грубим шрифтом терміни позначають напрямки розвитку науки програмування, що розпочався того часу у світовій практиці.

Як критерій для оцінки якості розширних систем програмування пропонується у статті Standish T.A.Extensability in Programming Langeage Design. – AFIPS Conference Proceedings, 1975, v.44, p.287-290.

розглядати наявність у ній засобів введення нових об'єктів трьох категорій*:*

1. Объектов, выразимых в терминах базисного языка и не требующих изменения базисного языка и его транслятора (синтаксические расширения: процедуры, макросы, конструкторы видов)

2. Объектов, ортогональных базисному языку, т.е. не выразимых в нем (семантические расширения). Добавления таких объектов в системах без специальных средств ее расширения выполняется либо с помощью "заплат" к транслятору с базисного языка, либо с помощью "заплат" к входной программе, написанных, как правило, в машинном языке (Ассемблере).

3. Объектов базисного языка, которые в процессе его развития меняют свое значение или интерпретацию в выходном языке. Это ведет к необходимости замены частей транслятора с базисного языка, а иногда и к полной его замене.

Средства для введения категории 1 присущи в той или иной степени всем языкам высокого уровня. А объекты категорий 2 и 3 реализуются сравнительно просто лишь в том случае, когда есть возможность изменять компилятор с базисного языка, т.е. когда компилятор написан по модульному принципу, управляется таблицами, запрограммирован на языке высокого уровня или, что еще лучше, реализуется с помощью компилятора компиляторов (Браун П. Макропроцессоры и мобильность программного обеспечения. М.: Мир, 1977.)

Саме така ідеологія незалежно втілена в РСП Т–М-220 (1967-1975 роки), в РСП Т-ЄС, Т-МВК тав РСП ТЕРЕМ (у 1980-их роках) на основі використання спеціальних підмножин мов для опису їх синтаксиса і семантики, що дозволило реалізувати в них об'єкти всіх трьох категорій, представлених на Схемі 1.

РСП ТЕРЕМ об'ємом 3260 операторів ПЛ/1 була прийнята у Фонд алгоритмів і програм (ФАП) АН УРСР (довідка № 329 за 7 грудня 1989 р.).

II. Реалізація мови-розширення СОСУД системи ПРОЕКТ-ЄС засобами РСП Т-ЄС

(1979- квітень 1980)

Автором транслятора з мови РПЛ/1 (ПЛ/1, розширеної операторами мови СОСУД), побудованого за допомогою РСП Т-ЄС є Щоголева Н.М.

У звіті відділу Теорії цифрових автоматів (ТЦА) за 1979 рік подається схема розширення мови ПЛ/1 операторами управління даними СОСУД системи ПРОЕКТ-ЄС:

Проектирование и реализация любой из подсистем математического обеспечения системы ПРОЕКТ-ЕС в языке ПЛ/1 основано на выборе подходящей системы программных модулей, реализованных в виде внешних процедур языка ПЛ/1. Связи между процедурами во время работы подсистемы осуществляются с помощью оператора вызова процедур языка ПЛ/1. Процедурное расширение языка ПЛ/1 можно определить как набор операторов, имеющих простой синтаксис, а в качестве семантики каждого из них – оператор вызова соответствующей процедуры языка ПЛ/1.

Реализация процедурных расширений языка ПЛ/1 для системы ПРОЕКТ-ЕС начинается с процедурного расширения СОСУД (операторы управления данными системы ПРОЕКТ-ЕС), широко применяемого в программах системы ПРОЕКТ-ЕС.

Розглянемо функції операторів розширення СОСУД зі статті авторів, які їх сформували.

В.В. Федюрко, О.Д. Фелижанко. "О методах реализации специализированных языков управления процессами функционирования систем программ" // Сб. тезисов докл. Всесоюз. семинара

Ґбавтоматизация производства пакетов прикладных программҐв, Таллин, 1980, с.72-76.

... Предполагается, что обрабатываемые данные имеют динамическую структуру, т.е. структуру, перестраиваемую в процессе обработки данных. Кроме того, данные имеют большой объем и не помещаются, как правило, в основной памяти. Средства управления данными должны обеспечивать возможность расчленения данных на части, например, путем их страничной организации. Эффективность доступа к данным со сложной динамической структурой обеспечивается совместным хранением элементов данных и информации о логических связях одних данных с другими элементами данных.

Рассматриваемая система управления данными включает средства, обеспечивающие формирование данных со сложной логической структурой и средствами обмена элементами данных между уровнями памяти. Они представлены в виде операторов специализированного языка программирования, расширяющих входной язык инструментальной ЭВМ. Основные возможности этого языка реализованы специализированной системой управления данными СОСУД.

Нижче в роздруківці сеансу роботи на ЕОМ ЄС, одержаній 28.05.1980 року під час демонстрації перед комісією, що приймала роботу, наведено приклад ПЛ/1-програми, де визначаються, вживаються і перекладаються мовою ПЛ/1 оператори мови СОСУД. Для зручності пояснення роздруківка програми та результатів розділена на фрагменти з нумерацією рядків. Номери рядків подаються в тексті грубим шрифтом.

Початок роздруківки експерименту, мета якого продемонструвати автоматизований перехід від операторів системи СОСУД до операторів мовою ПЛ/1

```
РАСШИРЯЮЩАЯСЯ СИСТЕМА ПРОГРАММИРОВАНИЯ Т-ЕС
ВХОДНАЯ ПРОГРАММА:

 1      СХЕМА(ОПЕРАТОР),2,M=INTER1.

        /* ЗАГРУЗКА СХЕМ ПРЕДЛОЖЕНИЯ БАЗИСНОГО ЯЗЫКА */
 2      СХЕМА(PROCEDURE),3,M=BASICL.
 3      СХЕМА(DECLARE),3,M=BASICL.
 4      СХЕМА(IF),7,M=BASICL.
 5      СХЕМА(),4,M=TRANS.
 6      СХЕМА(END),3,M=BASICL.
 7      СХЕМА(GO),3,M=BASICL.
 8      СХЕМА(GOTOY),3,M=BASICL.
 9      СХЕМА(DO),3,M=BASICL.

        /* ЗАГРУЗКА СХЕМ ПРЕДЛОЖЕНИЯ РАСШИРЕНИЯ СОСУД */

10      ОПЕРАТОР(ВЫДАТЬ СООБЩЕНИЕ()),11,Ф,1,M=SIMPLE1,T=(CALL ME8(#));.
11      ОПЕРАТОР(ЗАПИСАТЬ СТРАНИЦУ()),12,Ф,1,M=SIMPLE1,T=(CALL WRITS(#1));.
12      ОПЕРАТОР(ВЫЗВАТЬ СТРАНИЦУ(),()),13,Ф,2,M=SIMPLE1,
        T=(CALL REA4(#1,#2));.
13      ОПЕРАТОР(ПЕРЕВЕРНУТЬ СТРАНИЦУ(),()),14,Ф,2,M=SIMPLE1,T=(CALL REPLACS(#));.
14      ОПЕРАТОР(ВЫЗВАТЬ ПЕРВУЮ СТРАНИЦУ()),15,Ф,1,M=SIMPLE1,T=(CALL REFIRST(#));.
15      ОПЕРАТОР(ВЫДАТЬ СООБЩЕНИЕ () С БИТОВЫМ ПРОДОЛЖЕНИЕМ()),
        16,Ф,2,M=SIMPLE1,T=(CALL ME8BITS(#1,#2));.
16          LINEDEL:    PROCEDURE(K,P);
        /*  ВЫЧЕРКИВАНИЕ ЛИНЕЙНОЙ СВЯЗКИ ИЗ ПОРЦИИ;
                УКАЗАТЕЛЬ Р ОПРЕДЕЛЯЕТ
                ПЕРВУЮ ИЗ ВЫЧЕРКИВАЕМЫХ СТРАНИЦ.
                                                    */
17      DECLARE  K  DECIMAL  FIXED(1),
             (OLDKEY, KEY)  BIT(16);
18      DECLARE  I  BINARY FIXED(15),
             (YES, NO)  BIT(1)  EXTERNAL,
             T  BINARY  FIXED(16);
19      DECLARE  (FREE, LP, LINK)  BIT(16),
             (P, OKHO)  POINTER;
20      DECLARE   1 A   BASED(P),
                  3  E  BIT(8160),
                  3  (B, C(6), D)  BIT(16);
21      DECLARE F4A6(32) BIT(1) EXTERNAL,
             BITS  BIT(16),
             NUMB  PICTURE'9999';
22      ENTRY;   IF F4A6(30)  THEN  DO;
24           NUMB=88;
25      ВЫДАТЬ СООБЩЕНИЕ NUMB.
```

Фрагмент 1. Вводиться синтаксис базисної мови (рядки 2-9) та операторів розширення СОСУД (рядки 10-15) за допомогою ключових слів

Для прикладу у Фрагменті 1 припускається, що інформаційне забезпечення РСП Т-ЄС у початковому стані дозволяє їй розпізнавати лише одне ключове слово вхідної мови, а саме, СХЕМА. Далі за допомогою речень з цим ключовим словом вводяться інші ключові слова – базисної мови та її розширень.

***1**. За допомогою речення з ключовим словом СХЕМА вводиться ключове слово ОПЕРАТОР метамови Ld для опису синтаксису розширень. Процедура inter1, подана у цьому реченні, виконає занесення ключового слова ОПЕРАТОР у синтаксичні таблиці РСП Т-ЄС.*

***2-9**. Описані в дужках ключові слова базисної мови ПЛ/1: procedure, declare та інші, якими не можуть починатися оператори мови-розширення. З ключовими словами базисної мови пов'язується процедура BASICL, яка в режимі розширення базисної мови переносить оператори базисної мови з вхідного тексту в результат.*

***10-15**. Вводяться оператори розширення СОСУД. Для прикладу детально розглянемо*

*обробляння лише одного – першого оператора (рядок **10**):*
ОПЕРАТОР (ВЫДАТЬ СООБЩЕНИЕ ()),11, Ф, 1, М=symple1, Т= (call МЕ8(#);).

У перших круглих дужках вводиться синтаксис оператора розширення СОСУД, де "порожні" круглі дужки указують на позицію параметра, єдиного для даного оператора.
11 – номер нового оператора у списку таких операторів;
Ф, 1 – ознака того, що оператор має фіксоване число параметрів (один параметр);
М = symple1 – семантична процедура системи Т-ЄС, яка повинна обробляти всі оператори, де вводяться ключові слова мови-розширення. Головна її функція – занести ключові слова операторів розширення та їхні ознаки, в синтаксичні таблиці, а виклик процедури call МЕ8(#);) запам'ятати як семантику введенного оператора. В процесі трансляції програми оператор ВЫДАТЬ СООБЩЕНИЕ буде замінений викликом процедури МЕ8. На місці символа # буде змінна, значення якої потрібно видати.

***16**. Заголовок програми linedel, у якій використані введені оператори мови-розширення.*

```
24        NUMB=88;
25      ВЫДАТЬ СООБЩЕНИЕ NUMB,
26        END;
27        ОКНО=P;
28        KEY=C(K);  C(K)='0'B;  OLDKEY=B;
31      ЗАПИСАТЬ СТРАНИЦУ ОКНО,
32            IF NO THEN GO TO  WRECK;
34      ВЫЗВАТЬ СТРАНИЦУ KEY,ОКНО,
35            IF NO THEN GO TO CAN_NOT_TO_READ;
37       INCLUDE_TO_FREE:  T=1D;  FREE=A.B;
         /*    ОСВАЛОЖДЕНИЕ СТРАНИЦ СВЯЗКИ.      */
39       CHORA:  A.E='0'B; A.D=T;
41         DO I=2 TO 6;  A.C(I)='0'B;  END;
44      ПЕРЕВЕРНУТЬ СТРАНИЦУ'1,ОКНО,
45        IF YES THEN GOTO CHORA;
47        LP=A.B;    /*  АДР ЗАПИСИ В НЕЕ НОМЕРА
                  БЫВШЕЙ ПЕРВОЙ ЧИСТОЙ.  */
48       REGISTRATION:
         ВЫЗВАТЬ ПЕРВУЮ СТРАНИЦУ ОКНО,
49            IF NO THEN GO TO WITHOUT_FIRST;
         /*  РЕГИСТРАЦИЯ В СПИСКЕ СВОБОДНЫХ СТРАНИЦ  */
51        LINK=A.C(1);A.C(1)=FREE;
53      ЗАПИСАТЬ СТРАНИЦУ ОКНО,
54            IF NO THEN GO TO  WRECK;
56       LAST_PAGE:
         ВЫЗВАТЬ СТРАНИЦУ LP,ОКНО,
57            IF NO THEN GO TO WITHOUT_LAST;
59        A.C(1)=LINK;
60      ЗАПИСАТЬ СТРАНИЦУ ОКНО,
61            IF NO THEN GO TO  WRECK;
           /*  ВЫЗОВ ИСХОДНОЙ СТРАНИЦЫ. */
63        ВЫЗВАТЬ СТРАНИЦУ  OLDKEY,ОКНО,
64            IF NO THEN GO TO WITHOUT_OLD;
66          IF A.A0(30)='0'B THEN GO TO AWAY;
68          NUMB=87;  BITS=KEY;  GO TO SIGNAL;
71       WITHOUT_FIRST:  BITS='00000000000000001'B;
72          GO TO NOT_READED;
73       WITHOUT_OLD:  BITS=OLDKEY;  GO TO NOT_READED;
75       WITHOUT_LAST:  BITS=LP;  GO TO NOT_READED;
77       CAN_NOT_TO_READ:  BITS=KEY;
78       NOT_READED:  NUMB=86;  GO TO NO_1;
80       WRECK:  NUMB=82;  BITS=A;
82       NO_1:   NO='1'B;  YES='0'B;
```

Фрагмент 2. ПЛ/1 програма з операторами розширення СОСУД

Рядки Фрагмента 2:

***24**. Підготовка значення параметра NUMB для наступного оператора.*

***25**. Оператор ВЫДАТЬ СООБЩЕНИЕ, схему якого розглянуто вище.*

***31**, **34**, **44**, **48**, **53**, **56**, **60**, ... – інші оператори розширення СОСУД.*

Для виконання програм з операторами розширення СОСУД всі процедури, оператори викликів яких використані в програмі, повинні бути запрограмовані мовою ПЛ/1 зазделегідь і розміщені в базі даних РСП Т-ЄС. Таким чином, роль мови опису семантики розширень в РСП Т-ЄС виконує ПЛ/1 – базисна мова цієї системи.

```
82        NO_1:   NO='1'B;  YES='0'B;
84        SIGNAL:
          ВЫДАТЬ СООБЩЕНИЕ NUMB С БИТОВЫМ ПРОДОЛЖЕНИЕМ BITS,
85              AWAY;      END   LINEDEL;
86        K7.

   З А Г Р У З К А   С С П

СХЕМА,ОПРЕДЕЛЯЕМАЯ В ПРЕДЛОЖЕНИИ    1, ЗАНЕСЕНА В ССП СИСТЕМЫ  Т Е Р Е М  С ВНУТРЕННИМ НОМЕРОМ  2
СХЕМА,ОПРЕДЕЛЯЕМАЯ В ПРЕДЛОЖЕНИИ    2, ЗАНЕСЕНА В ССП СИСТЕМЫ  Т Е Р Е М  С ВНУТРЕННИМ НОМЕРОМ  3
СХЕМА,ОПРЕДЕЛЯЕМАЯ В ПРЕДЛОЖЕНИИ    3, ЗАНЕСЕНА В ССП СИСТЕМЫ  Т Е Р Е М  С ВНУТРЕННИМ НОМЕРОМ  4
СХЕМА,ОПРЕДЕЛЯЕМАЯ В ПРЕДЛОЖЕНИИ    4, ЗАНЕСЕНА В ССП СИСТЕМЫ  Т Е Р Е М  С ВНУТРЕННИМ НОМЕРОМ  5
СХЕМА,ОПРЕДЕЛЯЕМАЯ В ПРЕДЛОЖЕНИИ    5, ЗАНЕСЕНА В ССП СИСТЕМЫ  Т Е Р Е М  С ВНУТРЕННИМ НОМЕРОМ  6
СХЕМА,ОПРЕДЕЛЯЕМАЯ В ПРЕДЛОЖЕНИИ    6, ЗАНЕСЕНА В ССП СИСТЕМЫ  Т Е Р Е М  С ВНУТРЕННИМ НОМЕРОМ  7
СХЕМА,ОПРЕДЕЛЯЕМАЯ В ПРЕДЛОЖЕНИИ    7, ЗАНЕСЕНА В ССП СИСТЕМЫ  Т Е Р Е М  С ВНУТРЕННИМ НОМЕРОМ  8
СХЕМА,ОПРЕДЕЛЯЕМАЯ В ПРЕДЛОЖЕНИИ    8, ЗАНЕСЕНА В ССП СИСТЕМЫ  Т Е Р Е М  С ВНУТРЕННИМ НОМЕРОМ  9
СХЕМА,ОПРЕДЕЛЯЕМАЯ В ПРЕДЛОЖЕНИИ    9, ЗАНЕСЕНА В ССП СИСТЕМЫ  Т Е Р Е М  С ВНУТРЕННИМ НОМЕРОМ  10
СХЕМА,ОПРЕДЕЛЯЕМАЯ В ПРЕДЛОЖЕНИИ   10, ЗАНЕСЕНА В ССП СИСТЕМЫ  Т Е Р Е М  С ВНУТРЕННИМ НОМЕРОМ  11
СХЕМА,ОПРЕДЕЛЯЕМАЯ В ПРЕДЛОЖЕНИИ   11, ЗАНЕСЕНА В ССП СИСТЕМЫ  Т Е Р Е М  С ВНУТРЕННИМ НОМЕРОМ  12
СХЕМА,ОПРЕДЕЛЯЕМАЯ В ПРЕДЛОЖЕНИИ   12, ЗАНЕСЕНА В ССП СИСТЕМЫ  Т Е Р Е М  С ВНУТРЕННИМ НОМЕРОМ  13
СХЕМА,ОПРЕДЕЛЯЕМАЯ В ПРЕДЛОЖЕНИИ   13, ЗАНЕСЕНА В ССП СИСТЕМЫ  Т Е Р Е М  С ВНУТРЕННИМ НОМЕРОМ  14
СХЕМА,ОПРЕДЕЛЯЕМАЯ В ПРЕДЛОЖЕНИИ   14, ЗАНЕСЕНА В ССП СИСТЕМЫ  Т Е Р Е М  С ВНУТРЕННИМ НОМЕРОМ  15
СХЕМА,ОПРЕДЕЛЯЕМАЯ В ПРЕДЛОЖЕНИИ   15, ЗАНЕСЕНА В ССП СИСТЕМЫ  Т Е Р Е М  С ВНУТРЕННИМ НОМЕРОМ  16

В Ы Х О Д Н А Я   П Р О Г Р А М М А :

LINEDEL:    PROCEDURE(K,P)?
          DECLARE  K  DECIMAL  FIXED(1),     (OLDKEY, KEY)  BIT(16);
          DECLARE  I  BINARY FIXED(15),      (YES, NO)  BIT(1)  EXTERNAL,
```

***Фрагмент 3**. Повідомлення про занесення схем операторів розширення СОСУД в ССП (рос. мовою: Список Схем Предложений)*

Фрагмент 4. Результат трансляції процедури lindel

*У рядку Фрагмента 4 з міткою **ENTRY** оператор ВЫДАТЬ СООБЩЕНИЕ numb програмою symple1 замінений оператором виклику процедури: call me8 (numb). Решта операторів розширення СОСУД теж замінені на виклики відповідних процедур мови ПЛ/1.*
Реалізація розширення СОСУД – це лише початок складної роботи над створенням математичного забезпечення МВК з макроконвейєрною організацією обчислень.

III. Реалізація мов паралельного програмування сімейства МАЯК для МВК за допомогою РСП ТЕРЕМ

Основна задача Теремківського періоду-1 (1976-1990 років) – розробляння математичного забезпечення для многопроцесорного обчислювального комплексу (МВК) з макроконвейєрною організацією обчислень. МВК – це обчислювальна система з розподіленим управлінням, розподіленою пам'яттю та з універсальною системою зв'язків між процесорами.
У зв'язку з удосконаленням елементної бази ЕОМ відомі принципи фон Неймана про структурну і програмну організацію ЕОМ стали гальмом на шляху підвищення продуктивності процесу обчислень в ЕОМ. Перегляд принципів фон Неймана та формулювання нових розглянуто В.М. Глушковим із співавторами у статті:

Глушков В.М., Игнатьев М.В., Мясников В.А., Торгашев В.А. Рекурсивные машины и вычислительная техника. – Киев. – (Препринт / ИК АН УССР, 1974), 26 с.

Подано коротко зміст даної статті мовою оригіналу.

Авторы предлагают тип ЭВМ, соответствующий новым принципам, под названием рекурсивные вычислительные машины (РВМ), название которых отражает как структуру внутренних связей, так и структуру машинного языка...
Первый принцип организации РВМ заключается в произвольно высоком уровне машинного языка со сколь угодно большим числом языков низших уровней. В таком языке задаются рекурсивные правила перехода от элементов любого уровня (кроме низшего) к элементам предыдущего. При этом принцип центрального последовательного управления необходимо заменить принципом децентрализованного рекурсивно-параллельного управления вычислительным процессом...
Второй принцип заключается в том, что выполнению подлежат все программные элементы, для которых имеются в наличии операнды более высоких уровней, в состав которых входит данный оператор...
Третий принцип организации РВМ заключается в том, что структура памяти должна быть программно перестраиваемой с тем, чтобы в точности отразить структууру данных и программ, представленных на внутреннем языке. Такая организация памяти устраняет большинство проблем, связанных с динамическим распределением памяти...

Четвертый принцип организации РВМ заключается в принципиальном отсутствии ограничения на число элементов машин. Благодаря этому появляется возможность с единых конструктивных, технологических и программных позиций проектировать РВМ любого класса, начиная от малых машин и кончая суперсистемами и сетями вычислительных машин...

Пятый принцип организации РВМ заключается в гибкой программно-перестраиваемой структуре. Любая группа элементов РВМ, возможно, удаленных друг от друга, среди которых имеется хотя бы один процессор, который может в ходе выполнения программы образовать относительно независимый вычислитель, решающий свою задачу...

МВК з макроконвейєрною організацією обчислень слід розглядати як певний крок у напрямі комп'ютерних систем, що будуються на шляху відходу від принципів Фон Неймана, сформульованих ним у 1946 році під час побудови першого електронного комп'ютера ЕНІАК. МВК з макроконвейєрною організацією обчислень являв собою комплекс компонент (процесорів) різного призначення: арифметичний, керуючий, периферійний, комутаційний. МВК розроблявся в ІК АН УРСР у рамках кількох державних проблем-замовлень і за участю кількох організацій Радянського Союзу та підрозділів Інституту кібернетики. Зокрема, у відділі ТЦА велося розробляння системного математичного забезпечення рекурсивної машини: операційної системи та системи програмування.

Системне програмне забезпечення МВК розроблялося на основі мов програмування: сімейство мов МАЯК, стандартний ФОРТРАН, ПКОБОЛ – мова КОБОЛ, розширена засобами паралелізму, та РПЛ/1 – розширення мови ПЛ/1 операторами управління даними СОСУД. Транслятор для мови ПКОБОЛ розроблявся у відділі автоматизації програмування К.Л. Ющенко, решта трансляторів – у відділі ТЦА.

Одна із мов сімейства МАЯК, а саме, мова мультимодульного програмування (мова ММП) містила засоби динамічного управління ресурсами – процесорами, ініціювала обмін сповіщеннями між процесорами, містила засоби налагодження і використовувалася як мова системного програмування компонентів операційної системи. Трансляція програм мовою ММП здійснювалася в три етапи. Подаємо опис етапів за книгою

"Системное математическое обеспечение многопроцессорного вычислительного комплекса ЕС", коллектив авторов, Изд. ВВИА им. проф. Н.Е. Жуковского, 1986 г., 390 с.

Колектив авторів – це 41 співробітник, серед яких є всі, хто був причетним до створення математичного забезпечення МВК. Цитати взяті з написаних мною розділів книги.

11 На первом этапе выполняется декомпозиция мультимодульных программ на основную программу, которой приписывается тип УПР, и подчиненные программы типа АПР и УПР. Процесс декомпозиции включает:

– синтаксический анализ мультимодульной программы и выдачу сообщений о синтаксических ошибках;

– вычленение из мультимодульных программ текстов простых программных модулей;

– построение словаря объектов каждого простого модуля, описанных и/или употребляемых в нем. Для каждого простого модуля его словарь и словари всех охватывающих модулей образуют область видимости. В словарях имеется вся информация, которая необходима для независимой трансляции каждого модуля;

– сбор информации о мультимодульной программе и ее модулях для генерации управляющей программы группой процессоров и для генерации модуля взаимодействия.

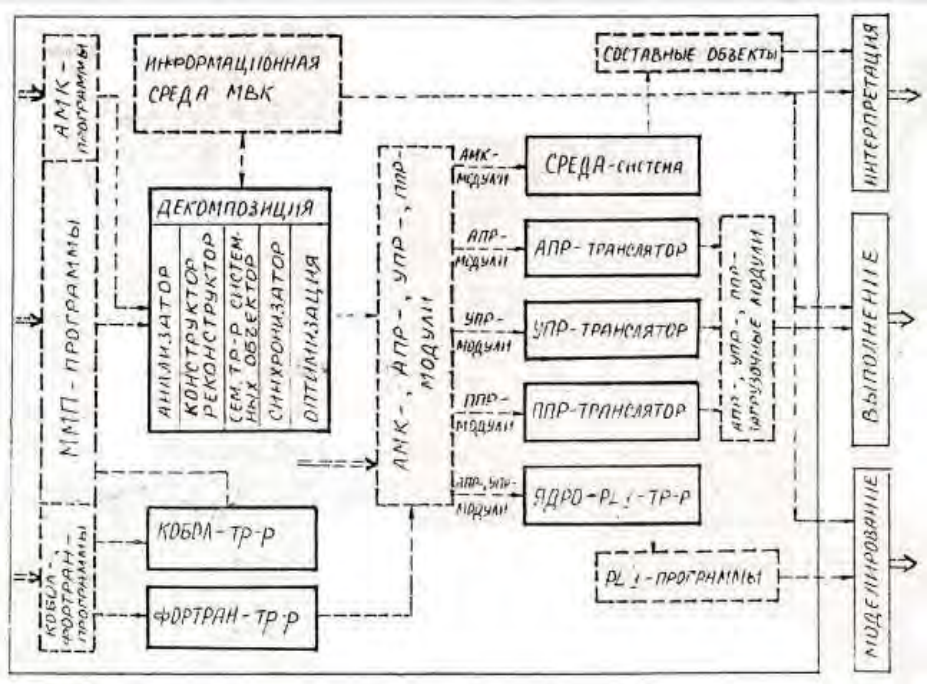


Схема 3. Архитектура системы параллельного программирования МВК

Второй этап – это раздельная и независимая трансляция УПР-, АПР- и ППР-модулей в языки, соответственно, АПР-0, УПР-0 и ППР-0, а также генерация в языке УПР-0 управляющей программы группой компонент (процессоров), в которой выполняется программа, и генерация в языке РПЛ/1 модуля взаимодействия мультимодульной программы с внешней памятью.

Первые два этапа – машинно-независимые, на втором этапе реализуется большинство распределяемых с ОС МВК функций, поэтому второй этап существенно зависит от решений, принятых в ОС МВК.

Третий этап – трансляция программных компонент на системном уровне языка МАЯК в языки различных процессоров (машинно-зависимый этап).

Поскольку промежуточным языком между вторым и третьим этапами является системный уровень языка МАЯК, то каждый транслятор третьего этапа может работать независимо от предыдущих и иметь самостоятельный вход.

Например, программы с жесткими требованиями к их эффективности могут быть написаны на системном уровне языка МАЯК (языке МАЯК-0) и протранслированы с помощью трансляторов третьего этапа.

Для розробляння Декомпозитора авторка цих спогадів використовувала РСП ТЕРЕМ, а побудову словника імен програмних об'єктів та змінних програмувала молодий спеціаліст випускниця КДУ ім. Т. Шевченка Валькевич Тетяна Арнольдівна.

УПР- та АПР-транслятори, розроблені з використанням РСП ТЕРЕМ у ролі компілятора компіляторів. Їх розробляли співробітники відділу ТЦА Кривий Сергій Лук'янович (УПР-транслятор) та Годлевський Олександр Богуславович (АПР-транслятор).

РПЛ/1 – мова ПЛ/1, розширена операторами виклику процедур управління даними. Система таких операторів мала назву СОСУД. Трансляція програм мовою РПЛ/1 на мову ПЛ/1 виконувалась РПЛ/1-транслятором, реалізованим Щоголевою Н. М. за допомогою РСП Т-ЄС. Отже, протягом Теремківського періоду я відповідала у відділі ТЦА за два проекти: перший – інструментальні засоби автоматизації програмування (РСП Т-ЄС та РСП ТЕРЕМ); другий – створення трансляторів за допомогою систем Т-ЄС і ТЕРЕМ та супроводження цих інструментальних систем під час розробляння трансляторів іншими співробітниками відділу. Ця робота давала результати для написання статей, доповідей тощо, завдяки чому можна дізнатися про завершення того чи іншого програмного проекта.

Статті та доповіді не були самоціллю. Відділ постійно виконував завдання різних державних

установ і організацій, що фінасували Інститут. Виконання завдань базувалося на програмуванні і закінчувалося написанням звітів та демонстрацією програмних засобів. Крім річних звітів про виконану роботу, співробітники і я в їх числі часто брали участь у семінарах, конференціях, симпозіумах, де були секції програмування. Написання доповідей спонукало до осмислення та підведення підсумків зробленого на ЕОМ, що разом з публікаціям и та звітами є основним джерелом даних про виконану роботу.

** * **

У програміста – варіації п'яти подій: алгоритм – програмування – робота на машині – пошук помилок – удосконалення програми (алгоритма), тобто, все спочатку. І так щодня, крім відряджень і відпусток. Часто вихідні дні були найрезультативнішими у цій роботі. Тому описувати процес програмування навіть помісячно неможливо. Тож спробую відновити пройдений шлях по роках.

*----------**1976** ----------*

***11 травня**. Починаючи з цієї дати після річної декретної відпустки, я почала працювати паралельно над такими завданнями:*

– перегляд систем программування з розширними вхідними мовами за попередні роки;

– вивчення матеріалів про машиниу ЄС 1060 та мову програмування ПЛ-1 у зв'язку з першочерговим завданням – перенесенням системи ПРОЕКТ на цю машину;

– участь у проектуванні та розроблянні математичного забезпечення для рекурсивної машини – операційної системи та системи програмування, які виконувалися у відділі ТЦА.

Перші результати виконання цього завдання – РСП Т з базисною мовою ПЛ/1 для машини серії ЄС (РСП Т-ЄС) були представлені у 1980 році на конференції у Талліні.

Отже, якщо перші два завдання з наведених вище – це перегляд і вивчення результатів роботи інших авторів, то третє вимагало інтенсивної творчої роботи протягом кількох років. До кінця 1976 року року було опрацьовано десятки статей про системи програмування з розширними вхідними мовами. Цікаві для нас роботи конспектувала. Але практично жодна з них не давала усіх відповідей на наші питання. Маємо йти власним шляхом. Почала працювати над системою РСП Т-ЄС (за моделлю РСП Т) для системи ПРОЕКТ-ЄС, де базисною мовою без вибору була мова ПЛ/1. У фірмі ІВМ була й мова системного програмування, яка не ввійшла до математичного забезпечення наших ЄС. Це підтверджує цитата з оглядової статті І.В. Поттосіна.

***И.В. Поттосин "Языки реализации для системного программирования". – Препринт/ Сиб. отделение АН СССР. ВЦ; Новосибирск, 1979. 24 с**.*

"ПЛ 360 имел развитые средства для управляющих структур и достаточно хорошее состветствие типов переменных форматам машинных представлений. Именно эти черты послужили той базой, на которой был создан ряд производственных инструментов СП для различных машин."

13

***30 травня**. Цього дня поставлена задача ознайомитися з наявними матеріалами, що стосувалися безпосередньо до наших завдань, зокрема, прочитати статтю В.М. Глушкова про рекурсивні машини; рекурсивні методи в програмуванні; мови Алгоритм, Структура, "Системне й теоретичне програмування-74" (праці конференції).*

Перше завдання для мене – описати систему програмування Т для М-220 як інтерпретатор на двох процесорах. 2 липня подала О.А. Летичевському звіт (20 сторінок з анотацією). Він мав дещо претензійну назву: "Многоступенчатая интерпретация языков высокого уровня на рекурсивных вычислительных машинах". Зберігаю копію звіту.

***23 червня** у відділі відбувся семінар (доповідач О.А. Летичевський), де було розглянуто три питання, пов'язані з новим проектом В.М. Глушкова побудови РВМ:*

1. Структура РВМ;

2. Мови паралельного програмування;

3. Операційна система.

Подаю конспективно основні пункти доповіді О.А. Летичевського (російською мовою), як початковий етап формування засад математичного забезпечення РВМ.

РВМ – многопроцессорная вычислительная система, содержащая не обязательно одинаковое пространство процессоров, которые могут взаимодействовать для обмена

информацией через две системы связи: локальную и глобальную.

Локальная связь. Предполагается, что на множестве процессоров задано отношение соседства – рефлексивное, симметричное. Примеры: линия, двумерная решетка, дерево. Любые два соседних процессора могут обмениваться быстро.

Глобальная связь: медленная (почта), быстрая (телефон).

Исходное состояние процессоров пассивное.

Виды режимов работы процессора:

– автономная работа;

– работа в режиме внешнего управления;

– в режиме массового обслуживания.

Автономная работа – выполнение работы через операционную систему (ОС), может находиться в состоянии ожидания, прерывание через ОС.

Внешнее управление – подчинение другому процессору через связь.

Методы решения задач:

а) автономное решение задачи одним процессором;

б) автономное решение задачи системой процессоров;

в) решение задачи асинхронно взаимодействующими процессорами с переменной структурой связи.

Информация об ОС для группы процессоров в режиме массового обслуживания.

Функции ОС: связь с периферией и установление новых связей.

Язык: структура языка верхнего уровня:

1. Обычные средства – алгоритмическая часть;

2. Описывать совокупность процессоров в языке СТРУКТУРА – для программирования связей в алгоритмах описания взаимодействующих процессоров.

3. Формировать связи, удалять старые.

***3 вересня** 1976 року. Семінар. О.А. Летичевський розглянув такі питання:*

1) Уточнение общих принципов (РВМ – Н.М.);

2) Язык программирования на РВМ;

3) Экспериментальное программирование, анализ динамических систем;

4) Многоступенчатая интерпретация многоуровневых языков.

Я почала працювати над перенесенням РСП Т на машину ЄС 1060, використовуючи мову ПЛ/1 як базову паралельно з опрацюванням накопиченої за рік моєї відсутності наукової літератури. У вересні 1976 року мене викликала Юлія Володимирівна Капітонова, заступник завідуючого відділом. Вона запропонувала мені бути інформатором у відділі з тематики відділу. Очевидно, знаючи, що я багато читаю, та маючи на увазі рік моєї відсутності на роботі, через що я могла втратити кваліфікацію програміста, вона знайшла для мене посильне навантаження. Саме по собі воно цікаве, я дійсно багато читала здебільшого зі свого вузького фаху: розширні мови програмування. Мене нова задача дуже засмутила, через що й запам'ятався епізод. У цій ситуації мене застав у кімнаті О.А. Летичевський і уточнив: займатися тим, чим я займалася і готувати статті за своєю тематикою.

***24-28 серпня**. Школа-семінар з проектування. Дані про цю школу не збереглися.*

***15 жовтня** виступала на семінарі у відділі з оглядом статей про розширні мови програмування.*

*----------**1977**----------.*

Основні пункти робочого плану відділу на 1977 рік, озвучені на семінарі Ю.В. Капітоновою.

I Построение математических средств проектирования;

II Создание и введение в эксплуатацию системы ПРОЕКТ-ЕС (Технический проект);

III Базовые алгоритмы проектирования;

IV Специальная операционная система.

Методика проектирования на языке ПЛ/1 ЕС: исследование проблем программирования, средства автоматизации проектирования. Очередные работы:

– три рукописи: дискретные преобразователи, логическое проектирование устройств, сборник по системам обработки математических текстов;

– конференция по автоматизации проектирования.

***27 січня** надійшла депеша: потрібно збирати документи для перевиборів на посаду молодшого наукового співробітника. Список необхідних документів довгий (рос.мовою): заявление с визой зав.отделом, листок по учету кадров, автобиография, отчет о научной работе (с визой зав.*

отд.), характеристика, список научных трудов, копии дипломов, аттестата (о высшем образовании, о присуждении ученой степени, ученого звания), выписку из протокола заседания партбюро.
Здається, досить було б заяви, звіту про наукову роботу та списку наукових праць. Решта є в у відділі кадрів. І це м.н.с. робив кожні 3 роки!

***11 березня 1977** року я затверджена у науковому званні молодшого наукового співробітника.*

***17 березня** віддала документи на продовження перебування на посаді молодшого наукового співробітника. Одним із 10 документів був звіт, який подаю:*

ОТЧЕТ

о проделанной работе мл. научного.сотрудника отдела № 100 М.Н.М. (1974-1977)

За отчетный период мною было выполнено:

1. Участие в разработке системы программирования системы ПРОЕКТ-2 для двухмашинного комплекса М-220 - БЭСМ-6. Результаты исследований отражены в отчете отдела №100 за 1974 г.

2. Участие в развитии и эксплуатации системы программирования системы ПРОЕКТ. Итоги этой работы оформлены в виде статьи и доклада на симпозиуме.

3. Реализация расширения языка программирования системы ПРОЕКТ, ориентированного на написание переводящих программ (расширение ТРАНСЛЯТОР).

4. Подготовка материала для предварительного отчета по математическому обеспечению многопроцессорных вычислительных систем.

5. Чтение лекций на школе ИК по системному проектированию и на всесоюзном семинаре по методам проектирования.

6. Реферирование статей по расширяющимся ЯП за период 1966-1976 годы. Часть результатов этой работы была доложена на семинаре отдела №100.

За отчетный период мною опубликовано 4 статьи, из которых 3 с соавторами.

22 марта 1977 г.

*У **березні** цього ж року підготовлена оглядова стаття про розширні мови програмування за період 1966-1977 роки. Частина цих результатів була оприлюднена у вигляді доповіді на семінарі відділу. Було розглянуто 11 робіт радянських авторів і 7 зарубіжних 15.*

Реферат з аналізом статей з нашої тематики подала 15 червня 1977 року О.А Летичевському.

Через деякий час я поновила текст реферату за рахунок нових статей. Ю.В. повернула мені реферат у 1980 році. Відтоді він зберігається у мене серед інших тогочасних раритетів.

***30 березня**. Семінар. Доповідач Погребинський С.С.*

Тема. Базовый внутренний язык процессора. Основні тези доповідача:

Тезис 1. Отношение времени отладки ко времени счета 1,5 к 1. Нужно учесть этот фактор.

Тезис 2. Стандартные программы, как правило, не используются, за исключением элементарных функций.

Тезис 3. Быстродействие растет с увеличением уровня языка.

Тезис 4. Выгоден и язык описания данных высокого уровня.

***6 квітня**. Семінар. А.А. Летичевский "О методе формализованных технических заданий": модели, методы, решение задач: анализ, синтез, оптимизация, структурирование объектов.*

***3 травня 1977** року затверджена в посаді молодшого наукового співробітника.*

***1 липня 1977 року** система РСП Т для М-220 була здана у РФАП.*

Мищенко Н.М., Федюрко В.В.,Фелижанко О.Д., Шерстобоева Г.К., Щеголева Н.Н. РСП Т на базе Автокода М-220 ИК АН УССР (Программы). (Часть 1, 2) // РФАП, справка №90 от 4.07.1977 Институт кибернетики АН УССР.

Літом працювала 6 днів у колгоспі, 1 день у вересні – в Ботанічному саду і 2 дні – на будові. Одна з робіт на будові стала останньою для мене. Того дня я прийшла на цю будову одна від Інституту. Мені було наказано перенести купу непотрібного паркету з 5-ого поверху на перший.

Перенісши його, я була б вільна піти з цієї роботи прямо додому. Звісно, я хотіла виконати її якнайшвидше, а тому набирала на ліву руку того паркету якнайбільше. Помітила, що йти вверх на 5-ий поверх значно легше ніж з паркетом униз. Зробила 13 ходів. Прийшовши додому, відчула біль у колінних зв'язках, коліна розпухли – мала чимало проблем, бо на роботу було

важко ходити із забинтованими колінами. Не піти на роботу не могла – не такі тоді були порядки, щоб хворіти без лікарняного.

Паралельно з описаними вище подіями складала алгоритми і програмувала РСП Т- ЄС.

***11 листопада**. Семінар. О.А. Летичевський "L2B: синтаксис и семантика".*

*З **19 по 23 грудня** у Києві відбулася Всесоюзна конференція "Автоматизация проектирования вычислительных машин", яку організовував Інститут кібернетики АН УРСР та Наукова рада з проблеми "Кібернетика" АН УРСР. Було подано біля 130 заявок на виступи. З колегами брала участь в організації конференції, за що одержали подяку:*

"Приказ о благодарности за проведение конференции № 87-К. 30 декабря 1977 г."

----------1978----------

***У січні** відбулося кілька семінарів, на яких заслухано доповіді:*

С.С. Гороховский, В.А. Бублик. Реализация L2-B (язык специальных структур данных).

В.В. Федюрко. Состояние языка СОСУД и ввод его в действие.

Обсуждение синтаксической подсистемы системы программирования.

***3 лютого.** Семінар у відділі: "Математическое обеспечение ПРОЕКТ-ЕС". Уточнення складу загальносистемного математичного забезпечення з різних точок зору:*

1. Штатные язык и операционные системы: ОС.4.0 и ОС.4.1, язык ПЛ-1, ЭКРАН.

2. СОС – специальная операционная система по идеологии ПРОЕКТ-1 (ЕС): система управления данными СОСУД, система управления директивами.

3. Система формирования структур данных: ввод-вывод текстов, составные объекты, системное управление лексическими единицами.

4. Интерактивная система обработки структур данных: редактирование текстов ИНЕСС, обработка составных объектов.

5. Интерпретирующая система обработки составных (КЛУБОК): формирование клубка составных, интерпретатор и, возможно, система программирования.

16

***20-22 березня** в Новосибірську відбувся Всесоюзний симпозіум "Перспективы развития в системном и теоретическом программировании", на якому виступив А.П. Єршов:*

"Сверхзадачей семинара является попытка увидеть вычислительное дело на рубеже столетий, очертить возможные достижения первого поколения исследователей в области программирования, показать перспективы и проблемы, открывающиеся перед научной сменой"

На симпозіумі розглянуті такі теми:

1. Математические основы системного программирования

2. Технология программирования

3. Методы синтеза, верификации и отладки программ

4. Языки программирования, методы их описания и реализации

5. Базы данных и системы управления ими

6. Архитектура вычислительных систем и программного обеспечения

7. Программное обеспечение сетей ЭВМ и систем коллективного пользования

8. Программное обеспечение многопроцессорных систем и параллельное прогр-ние

9. Новые средства общения с ЭВМ.

***20-23 червня** – відрядження до Москви. Юлія Володимирівна іноді посилала мене до Москви замість себе, коли її запрошували на семінари, нецікаві для неї. Я любила подорожувати, але з дітьми це не вдавалося, крім тих випадків, коли це потрібно на роботі. Це був такий випадок. Зараз уже й не пам'ятаю, у яку організацію того разу я їздила. Після поїздки написала лист москвичам з відділу В.М. Курочкіна, які мені 1973 року підписали відгук на дисертацію від ОЦ СРСР. З цього листа можна дізнатися про деякі події з мого життя. Після привітання я пишу:*

" Доклад, который я слушала, не потряс меня. Это не то направление, в котором мы работаем. Доклад относился к тому разряду автоматизации программирования, где задача формулируется на некотором языке (непроцедурном), система по этому описанию формирует программу из накопленных для данной проблемной области подпрограмм и выполняет ее. Языки для описания задач имеют общее название проблемно-ориентированных или языков весьма высокого уровня. У докладчиков таким языком является ОЕЯ (Ограниченный Естественный Язык). Очевидно, что в этом направлении авторы зашли весьма далеко, так как было сравнение с УТОПИСТОМ Тыугу не в пользу последнего.

В то время как всюду только и слышно об Утописте, а его создатель Тыугу на нем (в основном) защитил докторскую.

Москва, хоть и бегом я там была, оставила прекраснейшее впечатление. Я не первый раз в Москве, но я теперь так одичала, пребывая в основном в четырех стенах, что поездка оказалась настолько яркой, что я была как бы впервые в Москве.

А теперь о недавнем прошлом.

***28 февраля** (в последний день зимы) мне "удалось" поскользнуться и упасть в Институте на ступеньках на спину. Кости были целы, но много кровоизлияний, в том числе и внутренних. Лежала я пластом две недели... Эта история имеет длинное продолжение, но я больше об этом писать не буду. А вот с моим отчетом о расширяющихся СП было вот как. В прошлом году чуть раньше этой поры (15 червня 1977 р. – Н.М.) я отдала ее (работу) на прочтение своему начальству. В феврале этого года мне вдруг захотелось ее перечитать, стала я читать и, о ужас! вершина моего прошлогоднего творчества показалась мне такой никудышней, что я радовалась, что никому ее не послала. И я сразу бросилась переделывать. А это же, как говорится, в сводное от работы и "падений" время. Переделка затянулась, но уже выполнена, я не спешу печатать на машинке, т.к. выполняю другую работу, которая может повлиять на эту.*

Переделанная версия не претендует на обзор, в ней рассматриваются более или менее подробно лишь Ваша и моя работа, вернее, Ваша – менее, моя – более. Не умышленно, а просто, все, что я знаю. Естественно, свою я знаю лучше. Получилось совсем не то, что было, да еще и на 10 страниц меньше. Вот напечатаю и пришлю Вам, пока не стыдно при чтении. Ждать одобрения начальства не буду. Да и давать не буду, пока не сделаю новую работу – аналогичную систему на ЕС. Но над проектом еще надо работать.

Такое отношение моего начальства к моей теме объясняется тем, что сейчас главное – проектирование вычислительных машин и систем. Когда-то у меня была группа – 4 человека. Теперь я одна, все брошены на систему ПРОЕКТ, там основной акцент на ОС. В Институте есть отдел автоматизации программирования, то я иногда обращаюсь к ним. До сих пор удивляюсь, как мне удалось защититься. Оправданием этому служит, правда, то, что система работала и работает. И наши программисты отмечают многие ее преимущества по сравнению с тем, что они получили на ЕС.

Наша система на М-220 работала в нескольких режимах. Пополнение языка и выбрасывание ненужного тоже были автоматизированы. Это очень важно. Вряд ли я, одна теперь, сделаю такое на ЕС.

Мы на М-220 уже не работаем над развитием входного языка, но то, что там накопилось, используется и продается в разные организации. Мы потратили много сил, чтобы ее оформить в Фонд алгоритмов и программ, зато теперь систему легко передавать другим.

Лист закінчено побажанням колегам успіхів. Цікаво, що на ЄС ЕОМ я зробила навіть більше ніж на М-220. Можливо, тому, що не довелося програмувати транслятор з базисної мови, у ролі якої на ЄС ЕОМ була мова ПЛ/1 зі штатним транслятором.

Літо 1978 рік. З В.М. Глушковим у бібліотеці ІК. Ліворуч Н.М. Міщенко, праворуч А.Ю. Дорошенко і Ю.В. Капітонова

----------1979----------

Протягом року програмувала і налагоджувала РСП Т-ЄС для ЄС ЕОМ з базисою мовою ПЛ/1 одночасно з реалізацією розширення СОСУД. Пропоную фрагмент із звіту за листопад 1979 року за темою: "Вопросы проектирования и реализации алгоритмов в мультипроцессорных системах".uuuuuuПРОЕКТ-ЄС. Розділ 2.4.5:

2. Методы проектирования

2.4. Проектирование программ

2.4.5. Расширяющаяся система программирования Т-ЄС

Проектирование и реализация любой из подсистем математического обеспечения системы ПРОЕКТ-ЕС в языке ПЛ/1 основано на выборе подходящей системы программных модулей, реализованных в виде внешних процедур языка ПЛ/1. Связи между процедурами во время работы подсистемы осуществляются с помощью оператора вызова процедур языка ПЛ/1.

Процедурное расширение языка ПЛ/1 можно определить как набор операторов, имеющих простой синтаксис, а в качестве семантики каждого из них – оператор вызова соответствующей процедуры языка ПЛ/1.

Реализация процедурных расширений языка ПЛ/1 для системы ПРОЕКТ-ЕС начинается с процедурного расширения СОСУД (операторы управления данными), как наиболее широко применяемого в системе ПРОЕКТ-ЕС.

Анализатор схем, интерпретатор операторов расширения СОСУД, а также текстовый макропроцессор, осуществляющий подстановку фактических параметров в операторы CALL, находятся в стадии комплексной отладки.

Про реалізацію розширення СОСУД йшлося в Розділі II. Наведемо приклад дещо спрощеного опису лише одного оператора, а саме ЗАПИСАТЬ СТРАНИЦУ <номер страницы>. Описується схема і семантика такого оператора у вигляді речення, де місця параметрів позначаються парою порожніх дужок:

СХЕМА (ЗАПИСАТЬ СТРАНИЦУ ()), Т= (CALL WRIT5 ()).

Препроцесор, що обробляє цей опис, запам'ятає синтаксис – схему виклику процедури ЗАПИСАТЬ СТРАНИЦУ() та її семантику – CALL WRIT5 (). Процедура з назвою WRIT5() повинна бути запрограмована зазделегідь мовою ПЛ/1.

Розпізнання оператора зі схемою ЗАПИСАТЬ СТРАНИЦУ () ініціює звертання до процедури WRIT5 (). Наприклад, коли в програмі зустрінеться оператор ЗАПИСАТЬ СТРАНИЦУ Р, транслятор замінить його на оператор CALL WRIT5 (Р).

Реалізація процедурного розширення СОСУД була завершена до початку наступного 1980 року, про що було детально викладено вище у розділі II "Реалізація розширення СОСУД

засобами РСП Т-ЄС", де подана і роздруківка машинного експерименту для комісії, яка приймала нашу роботу.

----------1980----------

Цього року стався прорив у моїй праці! Завершено налагодження транслятора РСП Т-ЄС. Вперше після 4 років "мовчання" виступила на конференціях у Талліні та Ужгороді.
24 січня *на семінарі у відділі виступив О.А. Летичевський з доповіддю про математичне забезпечення (у тексті рос. – МО) РВМ (Проект окремого тому із структурою МО). Наводжу зміст цього тому як систему координат для розглянутих у подальшому тем. В дужках названо відповідальних виконавців у тих розділах, де я брала участь у програмуванні.*
Рабочий проект МО РВМ

Глава I. Функционирование РВМ.
Ґ 1. Технические средства
Ґ 2. Физическая память
Ґ 3 Организация информации
Ґ 4. Управление данными
Ґ 5. Организация вычислений

Глава II. Входные языки РВМ.
Ґ 1. Язык АМК
Ґ 2. Язык мультимодульного программирования (ММП)
Ґ 3. Язык для управляющих процессоров (УПР)
Ґ 4. Язык для арифметических процессоров (АПР)
mҐ 5. Языки для периферийного процессора (ППР): FORTRAN, ПЛ/1, КОБОЛ
Ґ 6. Языки спецпроцессоров
Ґ 7. Примеры программ для РВМ

III. Внутренние языки РВМ
Ґ 1. Внутренний язык УПР
Ґ 2. Язык микропрограммирования
Ґ 3. Интерпретация внутреннего языка УПР
Ґ 4. Внутренний язык АПР

Глава IV. Операционная система РВМ
Ґ 1. Управляющая программа и интерпретация языка директив
Ґ 2. Взаимодействие МВК с ППР
Ґ 3. Система управления данными
Ґ 4. Динамические распределения процессоров и контроль правильности функционирования РВМ

Глва V. Средства программ и данных для РВМ
Ґ 1. Организация работы пользоватерей на РВМ
Ґ 2. Система формирования данных
Ґ 3. Моделирование ММП-программ.
Ґ 4. Система конструирования программ
Ґ 5. Средства отладки
Ґ 6. Система обработки текстовой информации

Глава VI. Автоматизация проектирования схемного и программного оборудования РВМ
Ґ 1.Средства проектирования программ
Ґ 2. Расширяющаяся система программирования (Міщенко Н.М., Валькевич Т.А.)
Ґ 3. Системное моделирование РВМ
Ґ 4. Алгоритмическое моделировани РВМ
Ґ 5. Автоматизация микропрограммирования
Ґ 6. Автоматизация логического проектирования

Глава VII. Реализация входных языков РВМ
Ґ 1. Реализация языка АМК
Ґ 2. Трансляция основных языков РВМ (Берестовая С.Н. Валькевич Т.А., Годлевский А.Б, Кривий С.Л., Міщенко Н.М., Щеголева Н.Н. та ін.)
Ґ 3. Трансляция языка ФОРТРАН
Ґ 4. Использование языка ПЛ/1 (Щеголева Н.Н.)

Ґ 5. Использование языка КОБОЛ

Глава VIII. Математическое обеспечение спецпроцессоров

Глава IX. Прикладное математическое обеспечение РВМ.

Розглянемо детальніше підрозділи Глави 1.

Глава 1. Ґ 1. Технические средства (информация для пользователей)

Поданий вище запис в щоденнику, виконаний поспіхом під час семінару, свідчить: макроконвейєр – це АПР- і УПР-процесори та комутатор (обведені овальною лінією). Робота з макроконвейєром та з спецпроцесорами – через периферійний процесор (1045 або 1055).

Ґ 2. Физическая память (Физические данные): 256 блоков памяти. Блок – оперативная память одного процессора до 256 страниц, страница – 256 64-разрядных слов.

До 200 блоків пам'яті можуть бути блоками макроконвейєра та до 50 блоків – віртуальна пам'ять периферійної машини

Ґ 2. Организация информации.

Вся информация состоит из информатек. В информатеку помещаются связанные функционально программы и данные. Состав информации: динамические разделы для хранения структур данных (строк, составных объектов, информационных сетей), рабочие поля, библиотеки, именования, файлы.

Ґ 3. Управление данными

Три уровня буферизации: буфер памяти периферийного процессора (общесистемное средство), общая память макроконвейера, память одного процессора.

Ґ 4. Организация вычислений.

Общение с системой на языке директив. Возможны несколько потоков директив. Поток идет в УП периферийного процессора. Каждая директива выполняется одной программой. УП периферийного процессора обращается к УП макроконвейера. Происходит выделение ресурсов: одна или несколько программ периферийного процессора; определенное количество УПР- и/или АПР-процессоров.

***17 березня** поданий звіт про роботу м. н. с. Міщенко Н.М. за 3 роки (1977-1980):*

1. Сдана в РФАП РСП Т системы ПРОЕКТ. Объем системы программ 25726 машинных команд М-220. Выполнено совместно с В.В. Федюрко, О.Д. Фелижанко, Г.К. Шерстобоевой, Н.Н. Щеголевой.

2. Определена модификация РСП Т для реализации на машинах ЕС ЭВМ (РСП Т-ЕС).

3. Первая очередь системы Т-ЕС (программный базис) доведена на ЕС ЭВМ до опытной эксплуатации. Общая длина программ первой очереди около 1500 операторов языка ПЛ/1.

***У квітні** цього року із відділу ТЦА виділено групу співробітників у новий відділ Рекурсивних обчислювальних машин (РВМ), до якого приєднали групу програмістів на чолі з Берестовою С.М. з відділу програмування К.Л. Ющенко. Зав. відділом О.А. Летичевський. Я зарахована до нового відділу. Проте, працювали разом з відділом ТЦА, семінари теж були спільними.*

***8-10 вересня**. Відбулася Всесоюзная конференція "Автоматизация производства пакетов прикладных программ (Автоматизация производства трансляторов)". Таллин. Учредители: ВЦ АН СССР, ВЦ СО АН СССР, ИК АН ЭССР, Таллин, Политехнический институт.*

Секция 1. Технология. Председатель Ершов А.П., секретарь Мериste М.В.

Секция 2. Методы трансляции. Председатель Лавров С.С., секретарь Томбак М.

Секция 3. Теория. Председатель: Поттосин И.В., секретарь: Виллемс А.

Секция 4. Построение пакетов программ. Председатель Курочкин В.М., секр. Лийб.Д.

Секция 5. Реализованные СПТ. Председатель Редько В.Н., секретарь: Рохтла Х.

Міжсекційні доповіді:

1. Ершов А.П. Фундаментальные процессы трансляции

2 Лавров С.С. Язык ДЕКАРТ

3. Курочкин В.А., Серебряков В.А. Современные методы описания языков

4. Бежанова М.М., Тыугу Э.Х. Пути построения пакетов программ

5. Вооглайд А.О., Мериste М.В. Обзор систем построения трансляторов

На першій секції виступили (в дужках сторінки текстів у Збірнику тез):

Мищенко Н.М. "Расширение семантики входного языка расширяющейся системы

программирования ТЕРЕМ", с.29-32.

Щеголева Н.Н. "О погружении языков программирования и проектирования в вычислительную среду системы ПРОЕКТ", с.32-35.

Бублик В.В., Гороховский С.С., Чуйкевич В.С. Методы определения языков программирования для систем интерпретирующего типа, с.26-28.

На другій секції виступили:

Федюрко В.В., Фелижанко О.Д. О методах реализации специализированных языков управления процессами функционирования системы программ, с.72-76.

Зауваження. РСП, опис якої я подала на конференцію в Таллін, послуговувалася синтаксисом вхідних мов у вигляді множини ключових слів, якими починалися речення її вхідної мови. Наступного року ця РСП почала називатися РСП Т-ЕС аналогічно назві її прообразу РСП Т-М-220. А назва РСП ТЕРЕМ була перенесена на РСП для контекстно-вільних мов, яка почала розроблятися після подачі тез на конференцію в Таллін .

Две важные тенденции в развитии языков программирования – изменяемость и специализация – влекут за собой необходимость участия в их разработке программистов-пользователей. Расширяемые языки и системы – наиболее удобное средство для этого.

Основное достоинство расширяющихся языков с точки зрения пользователя состоит в том, что они освобождают от необходимости приспосабливаться к предлагаемым языкам, и дают возможность пользователю самому конструировать наиболее удобные средства программирования в его области применения ЭВМ.

Главный принцип реализации РСП ТЕРЕМ – введение расширений во входной язык системы с использованием языков высокого уровня для определения синтаксиса и семантики новых объектов. Развитие входного языка РСП ТЕРЕМ до языка высокого уровня на машинах ЕС осуществлялось в основном путем расширения семантики его объектов при сравнительно простом их синтаксисе, расширение которого происходит в пределах фиксированных классов синтаксических объектов.

Семантика языка-расширения – это система переводящих программ (модулей), ассоциируемых с синтаксическими объектами и образующих семантический транслятор (СТ) языка-расширения. В общей схеме трансляции программ в языке-расширении СТ выполняет лишь одну функцию: порождение или выбор одного перевода из нескольких, заранее заданных в языке более низкого уровня или в базисном языке. Остальные функции трансляции выполняют модули, общие для всех расширений входного языка.
Реализация первого языка-расширения служит апробацией методики построения расширений в системе Т-ЕС и превращает ее в систему построения трансляторов.

Доповідь була рекомендована до друку в ж. "Программирование", вийшла друком 1982 року.

Н.М. Мищенко "ТЕРЕМ – система построения расширяющихся систем программирования" ж. Программирование. – № 3. 1982. С. 57-63.

Фото В. Салмре. Вид на яхт-клуб Таллинского олимпийского центра парусного спорта

У Талліні нас дуже здивував порядок на вулицях та в готелі. Поїздка до Талліна відбулася невдовзі після Літніх Олімпійських ігор, які були проведені переважно у Москві, а в Прибалтиці відбувалася вітрильна регата. З такої нагоди у Талліні був побудований готель Ґболімпіяґв, у якому нас поселили на час конференції. У Талліні вперше були свідками, як вхідні двері до готелю перед гостем Ґбсамовідчиняютьсяґв. Зайшли у простору ліфтову кабіну, і через кілька секунд вона відкривається на вихід. Та ми ж не їхали! Настільки вона швидка і безшумна. У готелі сяюча чистотою кімната. У ванній для купання лежить папір з написом Disinfected. А на вулиці тільки ступиш однією ногою на мостову, як всі автомобілі зупиняються. Пізніше те ж саме я бачила закордоном. Цього не навчиш лише на час проведення Олімпіади.

Співробітник Інституту кібернетики АН ЕРСР Хен Лудвигович Салум провів для нас екскурсію по Старому Талліну. Хен Лудвигович працював деякий час у Києві в Інституті кібернетики, здається, на правах аспіранта, чи це так, уже не пам'ятаю. Він, до речі, самовільно організував відгук на автореферат моєї дисертації від ІК АН ЕРСР 1973 р.

***23-26 вересня**. Семінар "Проблемы реализации современных языков программирования". Ужгород. Учредители: Респ. дом экон. и н/т пропаганды общества "Знание" УССР, ИК АН УССР*

Ужгород. Міст через річку УЖ. Фото М. Плаксіна, 1970 рік

Виступала 24 вересня замість С.С. Гороховського та О.А. Летичевського за заявленою ними темою доповіді: "О реализации средств программирования для автоматизации проектирования вычислительных систем". Найбільше доповідачів було з України. Запам'ятався запах кави на вулицях та іноземна мова більшості пішоходів. Була організована екскурсія до Ужгородського замку. Вразила також залізниця до Ужгорода вподовж гори: гора з одного боку колій, глибока долина з населеними пунктами, як у прірві – з іншого.

***13 жовтня** у співаторстві з Гороховським С.С. послала до Москви в Інститут проблем управління тези доповіді на Всесоюзну нараду, яка мала відбутися у травні наступного року в Тбілісі. Нас не прийнялии у зв'язку з обмеженням на загальну кількість доповідей. Пізніше дізналася про учасників – всі високоповажні персони.*

Протягом року у відділі йшла інтенсивна робота по створенню керуючих програм та операційної истеми Макроконвейєра. Почала розробляти РСП ТЕРЕМ, орієнтовану на автоматизацію побудови трансляторів для мов, синтаксис яких описується формами Бекуса-Наура.

***6-13 листопада** лікувалася від запалення легенів у лікарні для вчених. Запам'ятався контингент у палаті:*

– актриса театру ім. Франка з її чіткою артикуляцією під час розмови;

– дружина знаменитого українського художника-графіка Георгія Якутовича, теж художниця, яка нічого не чула, але розуміла все, коли дивилася на того, хто говорить;

– жінка дуже похилого віку з міста Іванова, яка жила у Києві з сином і непривітною невісткою, передавала привіт Скурихіну В.І., якого вчила в Інституті в м. Іваново;

– жінка - екскурсовод по місту Києву.

Почула з перших вуст кілька цікавих історій з різних сфер творчого життя Києва.

----------1981----------

Робота року: програмування та налагодження модулів РСП ТЕРЕМ-ЄС .
***4 березня** відбулася нарада, де О.А. Летичевський навів повний список програм ОС та СП для Макроконвейєра і призначив виконавців. Я відповідаю за реалізацію мови ММП.*
***29 квітня.** О.А. Летичевський розповів про конференцію в Протвино, присвячену математичному забезпеченню машини ЕЛЬБРУС.*
Протягом травня відбулося кілька семінарів, на яких розглядалися кілька ММ-програм методів обчислювальної математики.

***8-11 вересня**. Семінар "Проблемно-ориентированные языки и специализированные системы". Респ. дом эконом. и н/т пропаганды общества "Знание" УССР, ИК АН УССР. Киев.*
Доповідь без друку Мищенко Н.М. "Расширяющаяся система программирования в системе ПРОЕКТ-ЕС". Зберегла запрошення з програмою семінару.
У кінці вересня мене обрали профоргом відділу за пропозицією Ю.В. Капітонової. Зараз дивно згадувати, скільки часу забирала профспілкова робота у багатьох людей. Вела щоденник профорга, тому можу для прикладу навести роботу новообраного профорга протягом першого після виборів місяця – жовтня.
1 жовтня. Здала профвнески з відомістю (їх же потрібно було збирати щомісяця).

***12 жовтня.** Потрібно виконати:*
1). Дати пропозиції щодо покращання умов праці
2). Завести журнал для інструкцій з техніки безпеки
3). Завести журнал оператовного контролю за станом охорони праці
4). Журнал-паспорт технічного стану і наявнасть засобів охорони праці
13 жовтня. Жалоба: Рисцову потрібна друкарська машинка, а до неї немає доступу
15 жовтня. Подати список дітей віком до 14 років з характеристиками здоров'я (а дітей тоді було до двох десятків).
19 жовтня. Пайки (масло, майонез, цукор)
Рисцов: посприяти в наданні йому кооперативної квартири (збирає підписи під заявою)
Роздала карточки учасникам профконференції ІК 22 жовтня
20 жовтня. Надійшла вимога до 5 листопада зібрати документи співробітників, яким потрібно санаторно-курортне лікування
До 15 листопада скласти список співробітників з характеристикою їхньої спортивної діяльності
21 жовтня. Подати список зіпсовапних меблів
22 жовтня. Профконференція тривала до 21 год. 30 хв. Присутність профорга обов'язкова
23 жовтня. Виробнича нарада:
Капітонова: 40 днів боргу у колгоспі, дисципліна, комісія, виставка;
Лябах: суботник;
Об'ява про санаторні путівки.
28 жовтня. Підготовка відомостей для збору членських внесків.
Це лише перший місяць моєї роботи профоргом і не найклопітнішиий. Хоч і не така це вже важка робота, якби не збивала з пантелику програміста, який і в сні шукає помилки.
Всі доручення, що надходили з профкому, намагалася виконувати відразу, щоб якнайшвидше звільнитися від клопоту. Були ще зобов'язання у профорга збирати внески в численні добровільні товариства у різний період року. У кожне товариство внесок був у межах 30 копійок. Я придумала оптимізацію цього процесу: умовила всіх співробітників зібрати один раз на рік 1 відсоток від платні. Коли надходила черга збирати внески в те чи інше товариство, я мала список наявних колег, одержувала підписи, а гроші у мене уже були. Таким чином платили всі, а не лише ті, хто був присутній у такий день. За це й побувала на Дошці пошани.
І так довелося працювати лише до вересня 1987 року. У вересні 1986 року, коли я в черговий раз понесла до місцевкому протокол зборів, на яких мене залишили на посаді профорга, замісник голови місцевкому, нова людина в ІК, запитав мене, чому не вибрали члена КПРС. У 1987 році профорга переобрали, але ним став знову не член КПРС.

***12 жовтня** відбулася нарада розробників РВМ.*

Секция системного матобезпечения

Клименко В.П.Внутреннее МО управляющего процессора

Бублик В.В. О языках программирования

Периферийный процессор: ЕС-1060, Макроконвейер: мультипроцессор ЕС-2701

Гороховский С.С. Управление функционированием МВК

Годлевский А.Б. О распараллеливании, ч. 1

Горлач С.П. Построение алгоритма распараллеливания ч. 2

*З **27 жовтня** на практиці в ІК Алла Корчевська (КДУ ім. Т. Шевченка), одна з небагатьох працьовитих і відповідальних студентів. До 13 листопада Алла швидко і якісно виконала завдання. Згодом виконала й дипломну роботу на "відмінно".*
А тим часом я закінчувала програмування і налагодження процедур РСП ТЕРЕМ, яка мала стати ядром Декомпозитора мультимодульних програм.

----------1982 ----------

Цей рік у нашій пам'яті обведений чорною рамкою: від нас пішов у безсмертя директор Інституту кібернетики АН УРСР Академік Віктор Михайлович Глушков. Відтоді Інститут носить його ім'я: Інститут кібернетики ім. В.М. Глушкова. 1982 рік – ювілейний: у грудні виповнилося 25 років Інституту і відділу ТЦА. Відзначення було відкладено на наступний рік і відбулося в мінорі. Потрібно сказати, що працюючи у відділі В.М. Глушкова, останні роки я зустрічалася з ним лише на велелюдних зібраннях в Актовій залі. Відомо, що Віктор Михайлович був зайнятий роботою мало не цілодобово. Ділові контакти з завідуючим відділом здійснювали його заступники по відділу, тож головні завдання відділу завжди були під його контролем.
За життя Ю.В. Капітонової щороку у відділах ТЦА та РВМ у серпні відбувався семінар, присвячений дню народження В.М. Глушкова, де виступали співробітники та гості зі споминами.

***16 квітня** відбулася нарада з РВМ. Розглядали двоє питань:*

1) у травні одержуємо "залізо" – тобто макроконвейєр;

2) стан справ з програмним забезпеченням. Перелік першочергових завдань, які стосувалися переважно програм управління (операційної системи).

***17 червня**. Підготовка звіту. Умовна назва звіту ПРОЕКТ-82.*

***21 вересня** нарада у відділі перед здачею теми: "Разработать математическую теорию построения многопроцессорных ЭВМ и создать экспериментальную систему для их исследования и моделирования". Потрібні такі документи:*
Звіт (1-2 примірники), відгуки, 4 акти впровадження, результати експериментів, публікації. Експерименти: приклади від авторів програм, приклади від перевіряючих, роздруківки. Методика випробувань системи, виписка з робочих планів, проект Акта про прийняття.
***4 жовтня** відбулося засідання міжвідомчої комісії, яка приймала звіт про виконання теми: ЃбРазработать математическую теорию построения многопроцессорных ЭВМ и создать экспериментальную систему для их исследования и моделированияЃв.*

Назва звіту "Экспериментальная система для исследования и моделирования многопроцессорных ЭВМ". Отчет в 2-х томах:

Том 1. Математическая теория проектирования многопроцессорных ЭВМ.

Том 2. Экспериментальная система для исследования и моделирования много процессорных ЭВМ.

2. Базовые средства проектирования многопроцессорных ЭВМ

2.3. Расширяющаяся система программирования ТЕРЕМ

3. Проектирование систем программирования.

Цього дня я та мої колеги виступали з доповідями перед комісією. Коротко тези мого виступу: базисні мови, засоби їх розширення, синтаксис і семантика вхідних мов, програмний базис, процес проектування РСП (в моєму архіві є текст виступу). Після виступів доповідачі відповідали на питання членів комісії. Подаю протокол засідання комісії мовою оригіналу.

П Р О Т О К О Л № 2

*заседания Межведомственной комиссии по приемке темы 0.80.16.09.07 **"Разработать математическую теорию построения многопроцессорных ЭВМ и создать экспериментальную систему для их исследования и моделирования" (4.10.82 г.,15.00)***

ПРИСУТСТВОВАЛИ: Мямлин А.Н.- председатель, Сергиенко И.В., Броев В.П., Дорожкин С.А., Задыхайло И.Б., Игнатьев М.Б., Капитонова Ю.В., Летичевский А.А., Казаков А.К., Рябов Г.Г., Чеботарев А.Н.- секретарь, Шигин А.Г., Андон Ф.И., Горлач С.П., Гороховский С.С., Мищенко Н.М.

1. СЛУШАЛИ: сообщение Горлача С.П. о моделировании архитектуры многопроцессорных ЭВМ (после вопросов через тире следуют ответы).

Вопросы:

1) Мямлин А.Н.: какие параметры характеризуют моделируемую схему решения задачи? – Горлач С.П. перечисляет параметры

2) Рябов Г.Г.: почему в качестве характеристики внешней памяти выбрана скорость обмена, а не, например, время доступа? – имеется и такая характеристика

3) Шигин А.Г.: нужна ли ретрансляция программы при изменении параметров? – ответ отрицательный

4) Мямлин А.Н.: что является результатом моделирования на втором этапе и каковы параметры моделирования на третьем этапе? – исчерпывающий ответ

5) Шигин А.Г.: предусматривается ли предварительное планирование эксперимента? – ответ утвердительный

6) Рябов Г.Г.: в каких пределах могут варьироваться параметры, откуда берутся ограничения на них? – ограничения на параметры вычисляются путем моделирования на других уровнях

7) Задыхайло И.Б.: каковы результаты моделирования конкретной системы с ЕС-1060 в качестве ППР? – при моделировании конкретные характеристики ЕС-1060 не учитывались

Замечания:

Рябов Г.Г.: задача ставится так: есть метод и есть класс структур, на которых он может быть реализован. Нужно выбрать архитектуру, на которой он реализуется наиболее эффективно.

Мямлин А.Н.: задачу вправе ставить и так, чтобы менять метод, а не архитектуру.

Вопросы:

8) Мямлин А.Н.: как предварительно оценить возможное влияние изменения параметров? – Горлач С.П. комментирует таблицы из демонстрационных материалов

9) Мямлин А.Н.: о времени, необходимом для вычисления эффективности на втором этапе моделирования – Ответ: десятки минут

10) Мямлин А.Н.: какова перспективность имеющейся системы для более сложных структур? – А.А. Летичевский: возможность моделирования достаточно сложных структур обусловлена многоуровневым моделированием

2. СЛУШАЛИ: сообщение С.С. Гороховского о моделировании программ операционной системы и прикладных программ

Вопросы:

1) Мямлин А.Н.: можно ли в результате моделирования проверить оптимизацию программ по каким-либо параметрам? – Да, на основании протоколов работы компонент

2) Матюхин А.Н.: какие параметры можно отменить? –перечисляет изменяемые параметры

3) Игнатьев М.Б.: операционная система сконцентрирована или распределена? Чем задается распределение? – Распределение задается на программном уровне

4) Игнатьев М.Б.: можно по-разному распределять операционную систему. В этом имеются возможности оптимизации. Рассматривались ли такие возможности? – Ответ положительный

5) Игнатьев М.Б.: исследовались ли вопросы надежности? Можно ли на системе промоделировать ситуации выхода из строя одного процессора? – В моделях ОС предусмотрены средства обеспечения надежного функционирования системы, для измерения же надежности нужны другие модели и системы

6) Задыхайло И.Б.: зачем моделируется выполнение программ вместе с операционной системой, почему нельзя отлаживать их с помощью отдельного эмулятора? – Поскольку некоторые ошибки можно обнаружить только при комплексной отладке.

3. СЛУШАЛИ: сообщение Мищенко Н.М. о проектировании систем программирования.

Вопросы:

1) Шигин А.Г.: как осуществляется оптимизация программ? – Завтра по этому вопросу будет

сообщение

2) Игнатьев М.Б.: в задачах имеется некоторый параллелизм. Есть ли возможность сохранить этот параллелизм при разработке программ? – Дает утвердительный ответ

3) Задыхайло И.Б.: до сих пор СПТ не позволяют получать хорошие трансляторы. Как вы надеетесь в более сложном случае получить обнадеживающие результаты? – Трудности возникают при создании универсальных СПТ, здесь же речь идет о разработке системы программирования, существенной частью которой является сама инструментальная система

4) Задыхайло И.Б.: почему недостаточна R-технология? – Инструментальная система ТЕРЕМ в большей степени, чем R-технология приспособлена для целей исследования языков программирования.

4. ДЕМОНСТРАЦИЯ экспериментов на машине по решению задач с помощью системы. Испытания прошли в полном объеме в соответствии с программой и методикой испытаний. Було підготовлено 2 експерименти.

Мета першого експерименту: продемонструвати синтаксичну підсистему СП для мови ММП сімейства МАЯК – мови мультимодульного програмування. Паралельно налагоджувалася РСП ТЕРЕМ для контексно-вільних мов, до яких належала і мова ММП.
Мета другого експерименту: реалізація розширення ППР сімейства МАЯК операторами мови СОСУД. Цей експеримент 1980 року наведено вище у Розділі II. "Реалізація розширення СОСУД засобами РСП Т-ЄС".

5. СЛУШАЛИ Мямлина А.Н. и Задыхайло И.Б. о задании по моделированию архитектуры многопроцессорных систем.
ПРЕДСЕДАТЕЛЬ доктор техн. наук . А.Н. МЯМЛИН
СЕКРЕТАРЬ канд. техн.наук А.Н. ЧЕБОТАРЕВ

***5-11 жовтня**. Школа-семинар "Параллельное программирование и высокопроизводительные системы", Алушта. Организаторы: Госкомитет по науке и технике СССР, Президиум АН СССР, Институт кибернетики им. В.М. Глушкова АН УССР, Симферопольский государственный университет.*

Тези доповідей (4 збірники-частини) вийшли друком у видавництві АН УРСР "Наукова думка".

***Часть 1. Формальные основы структурного параллельного программирования**. В этом сборнике представлена статья Ющенко Е.Л. "Теоретические и прикладные проблемы структурного и параллельного программирования".*

У доповіді К.Л. Ющенко простежена еволюція обчислювальних засобів у зв'язку з розширенням сфери застосування ЕОМ. Подаю тези доповіді К.Л. Ющенко мовою оригіналу.
Для решения сложных задач возникла необходимость пересмотра принципов, положенных в основу структурной и программной организации традиционных ЭВМ. Рассмотрены основные принципы, положенные в основу известных отечественных проектов перспективных высокопроизводительных мультипроцессоров.

1. Произвольно высокий уровень машинного языка.
2. Принцип децентрализованного параллельного управления вычислительным процессом.
3. Сочетание синхронной и асинхронной мультиобработки.
4. Модульно-иерархическая организация мультипроцессоров.
5. Потенциально неограниченное количество стандартизованных ресурсов.
6. Гибкая программная реконфигурация структуры мультипроцессоров.
7. Специализация памяти и самоидентификация данных.

Создание больших программных комплексов неразрывно связано с развитием технологии программирования (ТП) – совокупности знаний о способах и средствах разработки программ, оформившейся в качестве самостоятельной дисциплины к 1968 г., когда состоялась 1-я Международная конференция по ТП в ФРГ.
Известные отечественные технологические методы разработки программ:

– метод формализованных технических заданий (В.М. Глушков, Ю.В. Капитонова, А.А. Летичевский);
– Р-технология (И.В. Вельбицкий);
– композиционное программирование (В.Н. Редько);
– метод многоуровневого структурного проектирования программ (Г.Е. Цейтлин);

1983 рік. Учасники семінару колеги Анатолій Дорошенко (фотограф) та Сергій Горлач в Алушті

Часть 2. Средства параллельного программирования и их реализация. *У збірнику представлені 4 доповіді з відділу ТЦА, зокрема, доповідь:*

Ю.В. Капитонова, А.А. Летичевский, В.В. Бублик, С.С. Гороховский, Н.М. Мищенко "О реализации входных языков макроконвейерного вычислительного комплекса".

Макроконвейерный язык (МАЯК), созданный в Институте кибернетики АН УССР, представляет собой согласованное семейство языков программирования высокого уровня и предназначен для разработки последовательно-параллельных программ, выполняемых в многопроцессорном вычислительном комплексе (МВК) макроконвейерного типа.
У доповіді коротко представлені рівні мови МАЯК щодо виконуваних функцій. Виділяється мова мультимодульного програмування (ММП), трансляція якої має виконуватися в три етапи. Подається короткий опис кожного з них. Відзначається також важливість мови спілкування користувачів з МВК. Розглядаються передумови та засоби реалізації ММП-програм.
Часть 3. Методы параллельных вычислений и их сложность.
Часть 4. Организация вычислений на высокопроизводительных структурах.
Гороховский С.С., Капитонова Ю.В., Летичевский А.А., Федюрко В.В., Фелижанко О.Д., Щеголева Н.Н. "О разработке и реализации операционной системы МВК

.

1982 рік. Біля Чорного моря після доповіді...

20-22 листопада *в м. Протвино відбулося 4-е засідання Рабочої групи з реалізації мов програмування (абревіатури рос. мовою – РГ РЯП), що діяла при Комісії з системного математичного забезпечення (СМО) Координаційного комитету з обчислювальної техніки (ККВТ) АН СРСР. На засіданні розглядалися наступні питання:*

1. Модульний підхід до побудови трансляторів.

2. Системи побудови трансляторів.

3. План работи на 1983 рік.

У роботі засідання брав участь О.А. А.А. Летичевський, про що свідчать наступні цитати з протоколу 4-ого засідання РГ РЯП.

После прослушивания запланированных 11 докладов по теме заседания
А.А. Летичевский сделал сообщение о разрабатываемой в ИК АН УССР системе ТЕРЕМ...
Для рассмотрения системы ТЕРЕМ (она излагалась только в кратком сообщении А.А. Летичевского и вызвала общий интерес) было бы целесообразно поставить доклад о ней на следующем заседании РГ...
Был принят следующий план мероприятий на 1983 год:

1. Провести одно заседание РГ РЯП в г. Кишиневе с 3 по 5 июня, на котором рассмотреть:

– расширяемые языки и системы (отв. Д.Н. Тодорой и Л.Ф. Белоус);

– системное окружение языковых процессоров (отв. В.М. Пентковский)...

5. Подготовить создание целевой подгруппы (ЦПГ) по расширяемым языкам и системам (отв. Д.Н. Тодорой).
(Про ці плани Робочої групи у Протвино 1982 року я дізналася лише у 2010 р. з Інтернету).
Через деякий час О.А. Летичевський викликав мене і запропонував у нашій системі автоматизації програмування використати розвинутий синтаксис вхідної мови, а не такий, що послуговується списком схем операторів та описів, як це було в РСП Т для М-220. На що я відповіла, що в РСП ТЕРЕМ синтаксис вхідних мов описується контекстно-вільними
29
граматиками в Бекусово-Науровській формі. РСП ТЕРЕМ проходить експериментальну перевірку: на той час уже була реалізована синтаксична підсистема для ММП-транслятора.

2-4 грудня*. Всесоюзная конф."Современные проблемы кибернетики и вычислительной техники". Киев. Конференция посвящена 60-летию образования СССР.*

Учредители: Гос. комитет по науке и технике СССР, АН СССР, АН УССР, ІК АН УССР.
По вопросам математического обеспечения ЭВМ на конференции выступали:
БАБАЯН Б.А. Архитектура и ОС вычислительных комплексов, аппаратно ориентированных на ЯП высокого уровня

БУРЦЕВ В.С. МВК ЭЛЬБРУС. Выход на миллиардную производительность

ДОРОДНИЦЫН Анатолий Алексеевич Новые нетрадиционные применения математики и вычислительной техники

КОТОВ В.Е. Перспективы развития и реализации системы МАРС

МЕЛЬНИКОВ В. А. О разработке мульти процессорных систем

МИХАЛЕВИЧ В.С. Итоги и перспективы развития научных исследований в ИК им. В.М. Глушкова АН УССР

ПОСПЕЛОВ Г.С. Искусственный интеллект – новая информационная технология
САМАРСКИЙ А. А. Современные проблемы развития вычислительной математики

Організаційний комітет конференції склали співробітники відділів ТЦА та РВМ. Починаючи ліворуч: Ія Микитівна Коломойська, Надія Міщенко, Марина Байда, Тетяна Поліщук, Анатолій Чеботарьов, Ольга Феліжанко, Людмила Черкасова, ? Наталя Соболєва, Сергій Коляда

----------1983----------

Цей рік був багатий на результати досліджень і публікації. Насамперед була розглянута проблема розподілу функцій між операційною системою і системою паралельного програмування (СПП) МВК, забезпечення гнучкості СПП шляхом виділення універсальної частини СПП і, в разі необхідності, засобів її поступового розширення до конкретного транслятора. Такому підходу сприяла модульність СПП, табличний зв'язок між унівесальною складовою і додатковими модулями. Про це йшлося в публікаціях, де кожна публікація висвітлює одну-дві важливі характеристики програм згідно з темою відповідної конференції.

***22 січня** надіслала в Кишинів тези доповіді на тему "Исследование и реализация языков программирования в РСП Терем" на 4-ий Всесоюзний симпозіум "Системное и теоретическое программирование", який мав відбутися 31 травня – 2 червня 1983 р.*

***9 лютого** у відділі відбулася виробнича нарада. Основна тема – робочі плани на 1983 рік. Відділ є головним по країні щодо розробки математичного забезпечення МВК.*

***16 березня** подана у збірник ІК стаття у співавторстві з О.Б. Годлевським "Проблемы взаимодействия СП и ОС в многопроцессорных вычислительных системах"*
В статье проблема взаимодействия операционной системы (ОС) и системы программирования (СП) рассматривается как проблема оптимального распределения между этими системами реализуемых ими функций. Приводятся примеры реализации трех таких функций, характерных для многопроцессорного вычислительного комплекса с макроконвейерной обработкой данных. Отмечается важность совместного проектирования ОС и СП, необходимость сочетания гибкости и эффективности прохождения задач в вычислительных системах.

Збірник вийшов друком наступного 1984 року.

Годлевский А.Б., Мищенко Н.М. "Проблемы взаимодействия ОС и СП в многопроцессорных вычислительных системах". В кн. Развитие теории многопроцессорных систем. Сб. науч. тр. Киев: ИК АН УССР, 1984.

У цьому ж збірнику надрукована стаття колег:

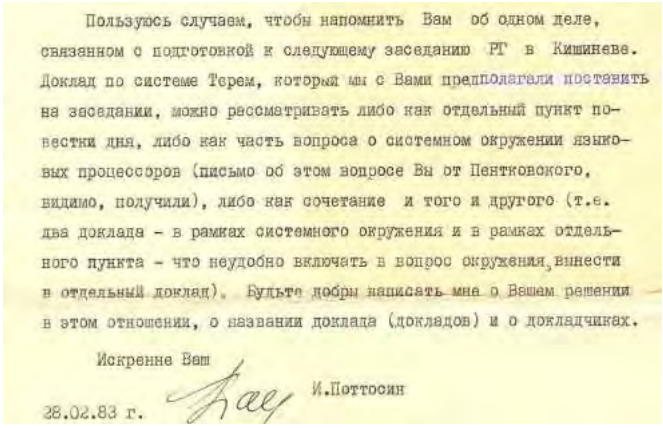
Федюрко В.В., Фелижанко О.Д., Щеголева Н.Н. "Средства обеспечения взаимодействия МВК с внешней средой".

В статье рассматриваются средства, обеспечивающие взаимодействие МВК с внешней

средой, которая включает специальным образом организованные программные модули. Описываются средства управления и перестройки информационной среды при решении сложных н/т задач. Приводится описание структур данных, используемых для организации внешней среды и средства работы с ними в пакетном и диалоговом режимах.

Перед 8 Березня. В Актовій залі Шура Момот Н.М. Міщенко, О.А. Летичевський, О.Д. Феліжанко, С.С. Гороховський, Н.С. Фурс

*На **початку березня** О.А. Летичевський одержав лист від І. В. Поттосіна такого змісту:*



Пользуюсь случаем, чтобы напомнить Вам об одном деле, связанном с подготовкой к следующему заседанию РГ в Кишиневе. Доклад по системе Терем, который мы с Вами предполагали поставить на заседании, можно рассматривать либо как отдельный пункт повестки дня, либо как часть вопроса о системном окружении языковых процессоров (письмо об этом вопросе Вы от Пентковского, видимо, получили), либо как сочетание и того и другого (т.е. два доклада - в рамках системного окружения и в рамках отдельного пункта - что неудобно включать в вопрос окружения, вынести в отдельный доклад). Будьте добры написать мне о Вашем решении в этом отношении, о названии доклада (докладов) и о докладчиках.

Искренне Вам

28.02.83 г. И.Поттосин

Після цього О.А. підійшов до мене з проханням написати назви доповідей, які ми маємо проголосити в Кишиневі на засіданні Робочої групи з реалізації мов програмування на початку червня 1983 року. Як відповідь на цей лист ми послали назви двох доповідей:
Ю.В. Капитонова, А.А.Летичевский, Н.М. Мищенко "Расширяющиеся языки программирования системы ПРОЕКТ и способы их реализации"

А.А.Летичевский, Н.М. Мищенко "Реализация языка МАЯК".

Невдовзі на ім'я Олександра Адольфовича прийшов другий лист, уже з Харкова від Леоніда Федоровича Белоуса, який після привітання пише:

На очередном заседании Рабочей группы по реализации языков программирования (РГ РЯП) комиссии по системному МО ККВТ АН СССР, которое состоится 3-4 июня этого года в г. Кишиневе по окончанию симпозиума СТП-83 (31.05 – 2.06), будет рассматриваться вопрос о расширяемых языках и системах.

Планируется следующая тематика:

1. Макропроцессоры и их применение

2. Расширемые языки и системы, способы их реализации

3. Механизмы расширения в СУБД и ППП

4. Мобильность программного обеспечения.

Если Вы предполагаете представить на РГ сообщение по этой тематике, пожалуйста, не позднее 31 марта вышлите материалы (содержание на 2-3 стр. и резюме) для предварительного рассмотрения по адресу: Харьков, Белоусу Леониду Федоровичу. Ответственные по данному вопросу: Д.Н. Тодорой и Л.Ф. Белоус.

Я написала назву, зміст робіт і резюме та послала їх 31 березня у Харків: "Расширяемые языки программирования системы ПРОЕКТ и способы их реализации"

РЕЗЮМЕ

Доклад представляет собой обзор расширяемых языков и систем программирования системы ПРОЕКТ, опыт создания и использования которых накапливался в течение более 15 лет. Отличительной особенностью рассматриваемых РСП является реализация средств СПТ, как наиболее мощных механизмов расширения. Средства расширения в РСП позволяют описывать реализуемый язык либо целиком, как в СПТ, либо по частям непосредственно во входных программах как в РСП.

***11-14 квітня**. Конференція "Автоматизация производства пакетов прикладних программ", Таллинн, ул. Эхитаяте тээ, 5. Таллинский политехнический институт, кафедра обработки информации.*

22 лютого послала тези доповіді "Об использовании практического подхода к построению трансляторов". Одержала відмову: "Сообщаем Вам, что в связи с ограниченными возможностями печатания, программный комитет, к сожалению, отклонил Ваши тезисы ..." Короткий зміст відхиленої доповіді. Суть используемого практического подхода к построению трансляторов состоит в том, что требуемый транслятор для некоторого языка программирования L реализуется путем специального расширения заранее заданного универсального программного базиса, являющегося общим в реализации всех языков класса, включающего и язык программирования L.

Поїхала без доповіді замість О.А. Летичевського на його прохання. Він був запрошений зробити пленарну доповідь, але не зміг поїхати. До Риги їхала сама, а звідти до Талліна з С.М. Берестовою. У Талліні нас зустрів Морозов С.І. Ми не домовлялися про зустріч – тим приємніше було, бо місце незнайоме і в темряві здалося безлюдним. Я себе на конференції почувала незручно. Організатори хоч і приховували незадоволення, але воно висіло в повітрі, я це відчувала. Дискомфорт передбачала, але ж була слухняна та ще й хотіла поїхати. Я весь час була заглиблена у програмування і конференції були для мене святом. Я була покарана: після такого демаршу мені відмовляли в участі у всіх наступних конференціях в Талліні.

***На початку травня** одержала запрошення на засідання Робочої групи з реалізації мов*

програмування, яке відбудеться у Кишиневі 3-5 червня 1983 року. Запрошення підписане А.П. Єршовим, який був на той час головою Комісії по системному математичному забезпеченню координаційного комітету з обчислювальної техніки (рос. абревіатура: СМО ККВТ) АН СРСР. Секретарем Комісії СМО була Бухштаб Діна Абрамівна, секретарем РГ РЯП Степанов Георгій Георгійович, який, зокрема, розмістив в Інтернеті детальний опис засідань Робочої групи у різних містах – єдине доступне тепер джерело відомостей про ці засідання.

1983 рік. Весняне прибирання придорожної полоси уподовж території Інституту кібернетики. На верхньому знімку, починаючи зліва біля багаття: Пятыгин С.А., Вольвач Ю.А., Колбасин Н.И., Соболева Н., Митченко А.И., Вершинин К.П., Горлач С.П. Фото Дорошенка А.Е.

На нижньому знімку: Федюрко В.В., Гороховский С.С. Фото Дорошенка А.Е.

31 травня – 2 червня. IV Всесоюзний симпозіум "Системное и теоретическое программирование" IV СТП-83. Кишинев. Учредители: АН СССР, Министерство высшего и среднего специального образования Молдавской ССР, Кишиневский Ордена Трудового Красного знамени Госуниверситет имени В.И. Ленина, Дом техники РС НТО.
Доповідь Н.М. Мищенко "Исследование и реализация языков программирования в РСП ТЕРЕМ".
При построении экспериментальной системы программирования (СП) для нового языка программирования (ЯП) необходимо, чтобы инструментальная система построения трансляторов (СПТ) содержала средства для исследования и развития ЯП в процессе его реализации и обеспечивала бы постепенный переход экспериментальной версии СП в производственную. Такой подход реализован в РСП ТЕРЕМ, используемой для экспериментальной реализации ЯП семейства МАЯК.
Исследовательский характер РСП ТЕРЕМ определяется оперативностью внесения изменений и дополнений в синтаксическое и семантическое описания реализуемого языка вплоть до их полной замены благодаря следующим свойствам системы:

– возможность описания синтаксических понятий непосредственно во входных программах, используя специальное подмножество базисного языкка РСП;

– модульная структура последовательно строящихся семантических трансляторов языка;

– гибкая связь модулей семантических трансляторов с соответствующими синтаксическими понятиями (программно-управляемая табличная связь).

Эти свойства способствуют усовершенствованию синтаксического и семантического описаний реализуемого языка и ообеспечивают постепенный переход СП от экспериментальной версии к производственной.

Описанный подход был применен при реализации некоторых языков семейства МАЯК для многопроцессорного вычислительного комплекса МВК.

3-5 червня відбулося засідання РГ РЯП. Наведу розклад лише тих засідань, на яких були заплановані наші доповіді та доповіді знайомих колег з інших організацій.

3 червня:

1. Дм.Бор. Подшивалов. О технологии в языках и методах трансляции. (40 мин.)
2. А.А. Летичевский, Н.М. Мищенко. Расширяющиеся языки программирования системы ПРОЕКТ и способы их реализации. (40 мин.)
3. С.С. Лавров. Расширяемость языков: теория и практика (40 мин.)
4. М. И. Селюн, Е.Н. Капустина. Система АБВ. (40 мин.)
5. Д.Н. Тодорой. Расширяемые языки и системы – способы их реализации. (40 мин.)
6. Г.С. Цейтин. Абстрактные типы данных и механизмы расширения в языках Alphard, CLU, ADA. (40 мин).

4 червня:

1. Ю.В. Капитонова, А.А. Летичевский, Н.М. Мищенко. Реализация языка МАЯК (1 час).

Перед днем доповіді я дуже хвилювалася і ніч перед виступом провела без сну. У день доповіді у мене був квиток на поїзд до Києва. Це було 3 червня. Коли купувала квитки, я не взяла до уваги розклад засідань Робочої групи. А тому купила квиток на Київ на той день, коли повинна була робити доповідь. Цікаво й те, що я взагалі не цікавилася ні розкладом занять, ні списком доповідачів на засіданнях Групи. Я про ці засідання нічого не знала і була впевнена, що виступала лише завдяки співавторству з О.А Летичевським.

На жаль, на той час я не володіла інформацією про цілі та задачі групи. Зробивши доповідь і навіть не чекаючи запитань, вийшла з аудиторії, а через кілька годин уже їхала до Києва. За моєї відсутності на засіданні я була введена в оргкомітет Робочої підгрупи з розширних мов програмування, про що мій керівник, який був на засіданнях, приїхавши до Києва, мені не повідомив. Сказав лише, що мене там похвалили і стартував підвищення мене в посаді: через

10 років після затвердження кандидатської дисертації я одержала посаду старшого наукового

співробітника (з 1 червня 1984 р.).

Повернувшись до Києва, я спочатку дивувалася, коли мені почали надсилати різні доручення

від голови підкомітету Тодороя Д.М., який в кінці-кінців "просвітив" мене. Здивувала байдужість керівника до уваги сторонніх, на яку заслужив підлеглий, адже це і його, керівника, заслуга.

Все те, що почули від мене слухачі на засіданні Робочої групи, я зробила одноосібно: алгоритми, програми. РСП ТЕРЕМ під час подання її у РФАП у 1988 році мала об'єм 3260 операторів мови ПЛ/1. А ще була система програмування ДЕКОМПОЗИТОР – складова СП для ММП сімейства МАЯК, ядром якої була РСП ТЕРЕМ, а доповнення до ядра – семантична підсистема об'ємом понад1500 операторів мови ПЛ/1. Певний час у постановках експериментів з Декомпозитором на ЕОМ мені допомагала О.А. Дудко, про що свідчить її співавторство у двох статтях. Потребувало часу і супроводження РСП ТЕРЕМ для реалізації трансляторів для трьох процесорів УПР, АПР і ППР, які розроблялися співробітниками відділу. А ще були студенти. Тож я постійно була у цейтноті

І не зважаючи на те, що я програмувала і супроводжувала складні програми трансляції та супроводу одноосібно, Дудко перевели на іншу роботу. До речі, у відділі директора, та й не тільки директора, кожний старший науковий співробітник офіційно мав помічника-програміста.

Одного разу мій науковий керівник О.А. Летичевський зацікавився, чому РСП ТЕРЕМ така велика. Удвох розглянули всі процедури. У них були реалізовані засоби СПТ, завдяки чому результат їх роботи не потребував постредагування на відміну від роботи на порядок меншої за об'ємом макротехніки, результат якої, одержаний після обробляння мов з типами змінних, може потребувати постредагування.

Та повернімося до виступу на засіданні Робочої групи. Насамперед процитую з Інтернету кілька фраз з протоколу засідань Робочої групи в Кишиневі, які стосуються моєї тематики.

В докладах и дискуссиях были обсуждены как общие проблемы расширяемых языков и систем, так и опыт конкретной реализации таких систем. Было отмечено разнообразие механизмов расширений, применяемых в языках и в СП. К средствам расширения можно отнести определение процедур, типов, макротехнику, введение абстрактных и инкапсулированных типов данных, средства синтеза программ...
В докладе Д,Н. Тодороя (Кишинев) предлагалась классификация средств расширения, методов их реализации, уровень включения расширения и общая модель расширяемой системы...

Необходимость объединения СПТ с расширяемой системой отмечалась в докладе А.А. Летичевского и Н.М. Мищенко (Киев), и возможность такого объединения демонстрировалась на примере РСП ТЕРЕМ. Авторы отмечали, что при создании базового языка с возможностями расширения минимальность базового языка может быть компенсирована мощными возможностями расширений...

На заседании Рабочей группы была отмечена важность работ по расширяющимся языкам и системам, принципиальная значимость средств расширения для языков и систем программирования, необходимость анализа направления ведущих работ. Было выделено два конкретных направления – анализ и классификация средств расширения в языках программирования и исследование характеристик существующих макрогенераторов.
Для совместной работы по этим направлениям было принято решение о целесообразности создания двух соответствующих целевых подгрупп (ЦПГ), в связи с чем ряду членов и наблюдателей РГ было дано задание подготовить к следующему заседанию РГ уточнение направлений деятельности и конкретные цели каждой ЦПГ.

Предполагаемый состав первой ЦПГ: Д.Н. Тодорой (председатель), Н.М. Мищенко и М.И. Селюн.Предполагаемый состав второй: Л.Ф. Белоус, М.С. Марголин, А.С. Марков, И.И. Пилецкий.

В качестве специального тематического доклада Рабочей группе был представлен доклад Ю.В. Капитоновой, А.А. Летичевского Н.М. Мищенко (ИК АН УССР), посвященный изложению методов реализации языка МАЯК, предназначенного для таких перспективных вычислительных средств как макроконвейерная вычислительная система.
На засіданнях було вирішено наступного року провести засідання Робочої групи у квітні-травні у Львові з основним питанням "Системное окружение языковых процессоров" і в Новосибірську з основним питанням "Методы реализации АТД в языках спецификаций".
Через деякий час на адресу авторів надійшов лист від Д.М. Тодороя такого змісту:
Решением РГ РЯП Ваш доклад на конференции рекомендован к печати в ж. "Прикладная информатика". Очередной номер журнала – тематический: Расширяемые средства программирования.

В связи с необходимостью своевременного сбора, рецензирования и окончательного оформления работ просим Вас содержание Вашего доклада оформлять в виде статьи с

учетом соответствующих требований журнала "Прикладная информатика" и выслать в наш адрес с приложением всех пополненных сопроводительных документов до 15 сентября 1983 года.

Адрес для переписки: 277003, г. Кишинев 3, ул. Садовая 60, Кафедра алгоритммческих. языков и программирования, д.ф.-м.н. Тодорою Д.Н. тел .раб. 25-00-21, доб.5-98.
Стаття була оформлена згідно з правилами журналу і вийшла друком 1984 року.
Ю.В. Капитонова, А.А. Летичевский, Н.М. Мищенко. Расширяющиеся языки и системы программирования системы "ПРОЕКТ". Сб. Прикладная информатика, Москва, "Финансы и статистика" 1984 (подписано к печати 7 августа), с. 163-171.

У статті були представлені: РСП РСП Т для М-220 і РСП ТЕРЕМ для ЄС ЕОМ.
РСП Т. Пропонується синтаксис та семантика вхідної мови, базисна мова АВТОКОД М-220 та її розширення ТРАНСЛЯТОР для побудови семантичних програм, технологічні особливості розробки системи Т, розширення вхідної мови: ЧАСТЬ – для роботи з частинами машинних слів, СИСТЕМА – для взаємодіїї програм над спільними даними, ТЕКСТ – для обробки символьних ланцюжків, ДЕРЕВО – для роботи з складними структурами даних.
На основі досвіду розробляння та використання системи Т в середині 1970-х років зроблено 5 висновків: про спільність семантичної бази мов-розширень; багаторівневий процес реалізації мов за допомогою системи Т; універсальність мовного і програмного базису; включення системи Т в систему ПРОЕКТ та доцільність розширення класу допустимих вхідних мов за рахунок мов, породжуваних контекстно-вільними граматиками, які описуються мовою форм Бекуса-Наура, відомою як мова БНФ.

РСП ТЕРЕМ. Розширний клас вхідних мов, граматики яких описуються мовою БНФ. Базисна
мова – параметр системи. Розглядаються різні варіанти вибору базисної мови, зокрема, вона може бути підмножиною мови системи ТЕРЕМ, або бути незалежною від неї. Від вибору базової мови залежить стратегія розширення. У статті розглядаються усі можливі випадки та відповідні засоби побудови мов-розширень. Подаються переваги машинно-орієнтовної мови у ролі базисної для забезпечення ефективності системного програмування.

Розглядається питання співвідношення між засобами РСП і СПТ.
Отже, РСП ТЕРЕМ – це універсальний базис, що містить:

1. Метамову LD для опису синтаксису.
2. КОНСТРУКТОР будує синтаксичні таблиці за описом синтаксису.
3. РЕКОНСТРУКТОР виконує генерацію опису синтаксису за синтаксичними таблицями.
4. Семантика (універсальні засоби перекладу).

РСП ТЕРЕМ реалізована на ЄС ЕОМ, базисна та інструментальна мова – ПЛ/1, об'єм базиса – понад 3000 операторів мови ПЛ/1.

Цього літа прийшла працювати у відділ випускниця Київського держуніверситету Валькевич Тетяна Арнольдівна, яка проходила у нашому відділі переддипломну практику і виконувала дипломну роботу. Наведу свій відгук на її дипломну роботу.
Рецензируемая дипломная работа посвящена разработке и использованию системы универсальных переводящих программ – макропроцессора, включаемого в состав расширяющейся системы программирования (РСП) ТЕРЕМ – инструмента для реализации СП языка МАЯК макроконвейерного вычислительного комплекса, разработка которого ведется в Институте кибернетики им. В.М. Глушкова.

Параллельно с работами над программами студентка Валькевич Т.А. изучила языки

программирования МАЯК и АДА. В дипломной работе приведены результаты сравнения средств этих языков и описан перевод фрагментов язка МАЯК в язык АДА с помощью разработанного макропроцессора. Выполнена автономная отладка макропроцессора и подготовлены примеры для комплексной отладки.

Студ. Валькевич Т.А. исполнительна, дисциплинирована, проявила хорошие знания языка ПЛ/1, освоила пакетный режим работы в ОС ЕС ЭВМ.

Работа заслуживает оценки ОТЛИЧНО.

Валькевич Т.А. розділила зі мною роботу над системою програмування для мови ММП сімейства МАЯК – вона розробила і запрограмувала словник ММП-програми.

Літо 1983 рік. Жукин. Крнференція по проектуванню.

На фото: ліворуч спиною до нас – Василь Федюрко, далі Надія Міщенко, Наталя Щоголева, Олександр Летичевський, Володимир Чуйкевич, Людмила Черкасова, Сергій Горлач.

***3-6 октября**. VI Всесоюзная школа-семинар "Параллельные вычислительные системы", **посвященная 60-летию академика В.М. Глушкова.** Учредители: н/т общество радиотехники, электроники и связи им. А.С. Попова, Центральное правление. Респ. правление АН УССР, Научный Совет АН УССР по проблеме "Кибернетика". Киев.*

На пленарных заседаниях выступили:

Ющенко Е.Л. Вклад В.М. Глушкова в теорию и практику программирования

Шура-Бура М.Р. Пути повышения уровня автоматизации программирования

Капитонова Ю.В., Летичевский А.А. О технологии управления параллельными вычислениями

Редько В.Н. Семантическое конструирование программ (основные результаты и открытые проблемы).

Вельбицкий И.В. Технология организации параллельной работы коллектива программистов

Цейтлин Г.Е. Математические основы структурного параллельного программирования

Анисимов А.В. Программирование параллельных процессов в управляющих пространствах

Погребинский С.Б. Обеспечение надежности многопроцессорных вычислительных комплексов.

В секции "Методы параллельных вычислений" были сделаны доклады:

Гороховский С.С. Языковые средства организации параллельных вычислений и их поддержка в операционной системе

Федюрко В.В., Фелижанко О.Д., Щеголева Н.Н. Средства обеспечения взаимодействия многопроцессорного комплекса с внешней средой

Мищенко Н.М. О средствах расширения в системах программирования

Тези останньої доповіді:

Средства расширения в языках и системах программирования делятся по своим возможностям на два класса:

– средства расширения синтаксиса языка (процедуры, конструкторы типов и др.);

– средства для расширения семантики (метасредства систем построения трансляторов).

РСП второго типа не очень много, так как проблема построения РСП такая же или даже труднее построения традиционной СПТ.

Для экспериментальной реализации нового языка нужна, с одной стороны, СПТ, которая позволила бы реализовать некоторый исходный вариант языка, а с другой стороны, последующая эволюция языка требует, чтобы результирующая СП была расширяемой. На практике это означает, например, возможность описывать новые конструкции языка непосредственно во входных программах в этом языке так, чтобы область действия этих описаний можно было ограничить этой программой или даже ее частью.

Таким двум требованиям удовлетворяет РСП со средствами СПТ в качестве механизмов расширения. В самом деле, средства СПТ в рамках РСП позволяют реализовать либо весь исходный вариант языка, либо некоторое его подмножество, после чего они (средства СПТ), оставаясь в рамках СП продолжают служить делу расширения (развития) реализованного варианта языка.

Принцип расширения, основанный на применении средств СПТ, у нас разрабатывается давно, и в настоящее время соответствующая РСП (РСП ТЕРЕМ) применяется для экспериментальной реализации языка МАЯК – языка мультимодульного программирования. Это не единственное возможное применение РСП ТЕРЕМ, в частности, и РСП вообще, основанных на средствах СПТ (например, CDL, расширение PL/1).

2 грудня я одержала лист від Д.М. Тодороя – керівника цільової підгрупи Робочої групи з реалізації мов програмування. На засіданні в Кишиневі на початку червня цього ж року було вирішено підготувати до засідання у Львові весною 1984 року матеріал, необхідний для затвердження цільової підгрупи з розширних мов програмування. У листі ставляться питання, на які потрібно знайти відповіді до початку роботи у Львові:

1. Цели и задачи ЦП по РЯ и С
2. Ближайшие задачи, подлежащие решению в области РЯиС (программа минимум)
3. Перспективы исследований в области РЯиС (программа максимум)
4. Как Вы себе представляете деятельность ЦП и Ваш возможный личный вклад в ее работу?
5. Какие, на Ваш взгляд, работы имеют наибольшее отношение к РЯиС?
6. Какие организации и/или исследователи и разработчики имеют наилучшие результаты в этой области?
7. Кого Вы можете привлечь к решению тех или иных вопросов в области РЯиС?
8. Какие другие вопросы могут быть решены ЦП РЯиС?

Моя відповідь на лист Д.М. Тодороя після привітання.

... Ваши вопросы очень сложные. Ответ на них, тем более полный, требует проведения специальных исследований. По моему мнению, некоторые Ваши вопросы, сформулированные в утвердительной форме, вполне могут войти в программу работы ЦП.

В итоге длительных размышлений над Вашим письмом я пришла к заключению:

1) хорошо было бы, если бы все, перечисленные Вами предполагаемые члены ЦП подумали над такими вопросами, внесли дополнения и обсудили бы это на ближайшем заседании группы; по-видимому, Вы это и предполагаете сделать;

2) работу следует начать с обзоров и библиографии РЯиС по нашей стране.

А теперь повторю те Ваши пункты, которые могут войти в план работы ЦП.

1. Качественное исследование расширителей, их типы и виды проделано в наболее общем виде в той статье, копию которой я Вам послала. Эту работу следует проделать сейчас на новых наших материалах и в более широком масштабе.

2. Следует провести исследования, какие отечественные работы следует относить к РЯиС. Другими словами, составить библиографию.

3. Лучшие результаты? Думаю, что "абсолютно лучших" нет. Есть лучшие в одном каком-то аспекте, в другом аспекте – лучшие другие. Как все в мире. Этот пункт должен быть основным принципом любого обзора по РЯиС.

4. Положив в основу работы программы–минимума ЦГ вышеперечисленные пункты, по ходу этой работы, я надеюсь, возникнут у нас и какие-то планы в отношении перспектив развития РЯиС и перспектив нашей работы.

Относительно деятельности ЦП. На первых порах, по-моему, следует действовать на фоне всей группы, то-есть, включать в программу РГ один-два доклада по РЯиС. Поскольку

я предполагаю, что это общие обзоры, то они могут представлять интерес для всех членов группы. Это – во-первых, а во-вторых, мы можем получить от членов группы замечания и советы, которые помогут становлению нашей программы.
Относительно моего личного вклада. Я лично, а может быть с А.А. Летичевским и Ю.В. Капитоновой, могу попытаться сделать доклад, сформулированный в первом пункте. Может быть, какие-нибудь мысли вызовут у Вас возражения, я готова выслушать.

Грудень 1983 рік. Кабінет зав.відділом №100. Наталя Щеголева, Валерій Гребнєв, Ольга Феліжанко, Василь Федюрко, Надія Міщенко, Костянтин Вершинін, Олександр Лялецький, Юлія Капітонова, Олександр Летичевський, Семен Гороховський, Олександр Годлевський.
І нарешті в кінці року довелося писати звіт про роботу у відділі за 1980-1983 роки.

ОТЧЕТ
о проделанной научной работе мл. науч. сотр. отдела № 105 М.Н.М. (1980-1983)
За отчетный период была проделана следующая работа:
1. Участие в разработке системы программирования языка МАЯК, которая является частью общесистемного МО многопроцессорной вычислительной системы. В частности, занималась разработкой структуры СП и отдельных компонентов этой структуры.
2. Развитие и сопровождение инструментальной системы проектирования трансляторов (система ТЕРЕМ) с целью применения ее для разработки компонентов СП МАЯК.
3. Сделаны научные доклады:
– на Всесоюзном семинаре "Автоматизация производства пакетов прикладных программ (автоматизация производства трансляторов)", (Таллин, 1980 год).
– на Республиканском семинаре "Проблемно-ориентированные языки и специализированные системы" (Киев, 1981 год).
– на 5-й Всесоюзной школе-семинаре "Параллельное программирование и высокопроизводительные системы" (Алушта, 1982 год).
– на IV Всесоюзном симпозиуме "Системное и теоретическое программрование" (Кишинев, 1983).
4. Принимала участие в написании отчета по теме "Разработать математическую теорию построения многопроцессорных ЭВМ и создать экспериментальную систему для их исследования и моделирования" (1982 год).
5. Опубликовала 2 статьи.
6. Руководила на общественных началах производственной и преддипломной практикой студентов факультета кибернетики КГУ им. Т.Г. Шевченко.

----------1984----------
Наведу назву теми з Робочого плану, яку ми виконували у 1984 році і здавали Держкомісії.

Проблема 080.14 *"Создать и освоить в производстве ЭВМ с производительностью до 10 млн оп/сек.*

Тема*: Разработать и сдать в опытную эксплуатацию ОС и систему параллельного программирования с языком высокого уровня для динамического распараллеливания в процессе выполнения программ на МВК с макроконвейерной организацией вычислений.*

Шифр темы: 0.80.14 (доп. тема)

Номер темы по плану Института С.Г.Д.100.05.

I. Разработать алгоритм функционирования программных средств и осуществить их экспериментальную проверку на комплексе ЕС 1060 и ЕС 2701.

1. Разработать языковые средства параллельного программирования, общую схему и алгоритмы управления процессом вычислений в МВК с макроконвейерной организацией вычислений. I кв. 1984 г. Исполнители: с.н.с Гороховский С.С., рук.гр. Федюрко В.В., инж. Крат С.П., зав. отд. Молчанов И.Н.

2. Разработать алгоритм анализа, преобразования и описаний параллельных программ с целью генерации компонент ОС и программных модулей, составляющих среду исполнения программ. II кв. 1984 года. Исполнители: м.н.с. Мищенко Н.М., м.н.с. Годлевский А.Б., зав. отд. Ющенко Е.Л.

3. Разработать моделирующие средства и провести моделирование алгоритмов ОС. III кв. 1984 г. Исполнители: инж. Горлач С.П., инж. Морозов С.И., и.о. зав.отд. Паньшин, с.н.с. Гребнев В.А.

4. Разработать экспериментальный комплекс системы параллельного программирования. ***IV кв. 1984*** *г. Исполнители: м.н.с. Мищенко Н.М., м.н.с. Щеголева Н.Н., с.н.с. Берестовая С.Н., зав. отд. Молчанов И.Н.*

II. Разработка технического проекта и экспериментального образца процессора макроконвейерной обработки данных (х/д 220-81 с п/я М-5769), (ЕС 2701)*.*

III Разработка методов и средств обеспечения надежности проектирования дискретных устройств (х/д №334-83, п/я А-1183).

О ж и д а е м ы е р е з у л ь т а т ы: Будут разработаны общие схемы и алгоритмы функционирования ОС, СПП и проведена их экспериментальная проверка в процессе решения задач на комплексе ЕС 1060, ЕС 2701.

Выполняется по постановлению ГКНТ СССР от 15.08.1983 г. №445.

Начало III квартал 1983 года. Окончание IV квартал 1985 года.

Затраты на весь срок 2700 тыс. руб. На 1984 год 1350 тыс. рублей.

Количество научных сотрудников 15, вспомогательных – 8.

Отделы 100, 105, 145, 150, 430, СКБ, СКТБ.

Научный руководитель темы академик А.С. Михалевич.

Заместители руководителя Ю.В. Капитонова, А.А. Летичевский.

З перших днів 1984 року розпочалася активна підготовка до випробування макроконвейєрного машинного комплексу. У відділі ТЦА щотижня проводилися оперативки, визначалися першочергові задачі для програмування математичного забезпечення та його належного функціонування на макроконвейєрному комплексі. Дійшло до того, що я не змогла поїхати у травні до Львова на засідання Робочої групи з мов програмування, не дивлячись на те, що там мала бути затверджена цільова підгрупа з розширних мов програмування, а я входила в оргкомітет цієї підгрупи. Та був у мене й особистий стимул – у Львові жила родичка, моя тьотя, поїздка до якої була завжди святом. Та ще й Львів був і залишається моїм любимим містом...

18 червня *я одержала лист від Тодороя Д. М. з Кишинева:*

Продовження листа. *В связи с этим прошу Вас (1) рассмотреть представленный список литературы, уточнить его, дополнить, отпечатать и выслать в мой адрес в 3-х экземплярах. Дальнейшее размножение примерно 100 экземпляров я беру на себя.*

В связи с подготовкой рабочего семинара 35-40 человек (школы, совещания) по РСП прошу Вас выслать в мой адрес:

2.1. Список тем, подлежащих рассмотрению на семинаре

2.2. Предпочтительные места встречи и время проведения семинара

2.3. Список заинтересованных организаций, подлежащих охвату, их адреса, телефоны,

конкретные лица

2.4. С каким докладом предполагаете Вы и Ваши коллеги (сотрудники, заинтересованные лица) выступить на этой встрече. Я должен обмозговать поступившую от Вас информацию (и от остальных членов ЦП) до конца июня сего года и сообщить наше решение Комиссии в начале июля сего года.

Поэтому прошу Вас по возможности срочно ответить на мое письмо с рядом уточнений и конкретных предложений.

P.S. Очередное заседание РГ РЯП: 29.10.84 – 3.11.84, г. Новосибирск.
Следующее заседание РГ РЯП: апрель 1985 г., г. Баку (в горах).

Очень прошу Вас ответить. Ваш Дм. Тодорой

Моя відповідь Дмитру Миколайовичу:

Многоуважаемый Дмитрий Николаевич!
К сожалению, я вынуждена извиниться перед Вами за задержку ответа на Ваше письмо. Получила я его 18 июня, но с тех пор так и не смогла взяться за работу над списком. У меня сейчас чрезвычайно трудный период, 30 июля начинаются госиспытания РВМ, в котором я ответственный исполнитель РСП (расширяющейся системы программирования). Оргработа, внедрение, РСП, создание документов – вне всякой очереди. Поэтому у меня нет выходных и не будет отпуска, поэтому я не была во Львове. Госиспытания и сдача опытного образца будут длиться месяца 2-3, поэтому вряд ли я буду в Новосибирске.

А теперь о деле.

1. Список хотя и полный, но устаревший, я знаю, что нужно выбросить, но не могу его дополнить. Принцип сокращения: до 1978 года только обзоры и постановочные статьи. Например, конференция в Тбилиси, Ахо и Ульман. Предлагаю отсрочить рассылку, нельзя в таком виде. Буду над ним работать.

2. В связи с подготовкой – то, что я писала раньше.

3. Где проводить заседания – мне всеравно.

4. Спасибо за монографию. Мои последние работы – в Трудах Кишиневского симпозиума и в "Прикладной информатике".
Вышлю монографию, которую напишу.

На цій оптимістичній фразі я закінчила лист.

***27 липня** на засіданні вченої ради відділення ТЕК (теоретичної і економічної кібернетики) була затверджена тема кандидатської дисертації моєї найближчої колеги того часу Валькевич Тетяни Арнольдівни. Валькевич Т.А. працювала в ІК з 9 вересня 1983 року після закінчення Київського держуніверситету ім. Т. Шевченка. Тоді тему дисертації назвали "Проблемы семантической трансляции параллельных языков".*

Захист дисертації відбувся в 1994 році. Тема дисертації трансформувалася у таку:

*"Инструментальные средства поддержки обработки онтологической информации про**граммных структур многокомпонентного программирования".***

***Комісія з приймання роботи по Макроконвейєру почала працювати 30 липня і працювала біля** 3-х місяців. Мені було особливо складно через те, що я відповідала персонально за значну частину математичного забезпечення. Сторінки щоденника рясніють записами про підготовку і закінчуються 31 липня. Перебіг самих випробовувань я не фіксувала.*
Першим записом у щоденнику після випробувань є запис висновку комісії:

ПРИКАЗ

4 октября 1984 г., Москва

В соответствии с решением Комиссии Президиума Совета Министров СССР от 10.07.81 № 233, постановлением ГКНТ и Комиссии Президиума Совмина СССР от 20.11.81 №442/377 (приказы Минрадиопрома от 11.08.81 № 467 и от 16.03.82 № 139, соответственно) Институтом кибернетики им. В.М. Глушкова АН УССР совместно с НИИ центром электронной вычислительной техники и Пензенским заводом "ВЭМ" выполнены опытно-конструкторские работы по разработке и изготовлению опытного образца процессора макроконвейерной обработки данных (ЕС 2701) и опытного образца специализированного процессора интерпретации языков высокого уровня (ЕС 2680).

Работы выполнены полностью, испытания прошли успешно и предъявлены на

государственное (межведомственное) испытание.
Це були випробування техніки, які можна було провести лише роботою на ній математичного забезпечення (МО). А випробування ж самого МО відбувалося наступного 1985 року.

***18-19 вересня**. **Семінар "Проблемы организации взаимодействия "человек-ЭВМ"**. Респ. Дом экономической и н/т пропаганды общества "Знание" УССР, ИК АН УССР, Киевское городское правление НТО РЭС им. А.С. Попова. Киев.*

Доповідь без друку "Система программирования МВК".

***24 жовтня**. Семинар Научного совета АН УССР по проблеме "Кибернетика". Секция 1. Семинар 1.1. Теория автоматов и ее применения. Прослухано дві доповіді.*

Перша:. Н.М. Мищенко "О реализации семантических подсистем трансляторов в РСП ТЕРЕМ". В докладе описан процедурный подход к реализации семантических подсистем трансляторов, основанный на представлении процессов перевода в виде суперпозиции семантических действий, нагруженных на грамматику реализуемого языка. Определена информационная и программная связь синтаксических подсистем трансляторов с системами процедур, выполняющих семантические действия. РСП ТЕРЕМ используется для построения нескольких трансляторов, входящих в состав математического обеспечения МВС с макроконвейерной организацией вычислений.

*За матеріалом доповіді **28 березня 1985 року** підписана до друку стаття:*

***Мищенко Н. М. О реализации семантических подсистем трансляторов в РСП ТЕРЕМ.** – В кн.: Автоматизация проектирования многопроцессорных вычислительных систем: Сб. науч. тр. Киев: ИК АН УССР, 1985.*

Друга: Т.А. Валькевич, Е.А. Дудко, Н.М. Мищенко "Декомпозиция мультимодульных программ". В докладе представлены функции и результаты первого этапа обработки мультимодульных программ в языке МАЯК, целью которого является расчленение мультимодульной программы на отдельные модули и обеспечение дальнейшей раздельной и независимой трансляции этих модулей в системе программирования МАЯК. Обсуждается выходная структура данных этапа декомпозиции, отражающая структуру входной мультимодульной программы и содержащая в качестве основных подструктур словари составляющих ее модулей.

*За матеріалом доповіді 6 листопада **1985 року** підписана до друку стаття:*

Т.А. Валькевич, Е.А. Дудко, Н.М. Мищенко "Декомпозиция мультимодульных программ". – В кн.: Организация вычислений в многопроцессорных ЭВМ: Сб. науч. тр. Киев: ИК АН УССР, 1895.

***Грудень.** Вийшла друком стаття*

Мищенко Н.М. О реализации синтаксических подсистем трансляторов в РСП ТЕРЕМ // В кн.: Методы и средства проектирования дискретных систем Сб.науч. тр. Киев: ИК АН УССР, 1984, с. 16-23.

В РСП Терем предложен практический подход к построению синтаксических подсистем трансляторов, основанный на использовании средств построения трансляторов. К таким средствам относится синтаксическая подсистема РСП ТЕРЕМ, содержащая Анализатор, управляемый синтаксическими таблицами, и Конструктор, генерирующий эти таблицы по описанию грамматик в метаязыке, близком к языку БНФ. Описаны отличия матаязыка для описания грамматик от языка БНФ.

Короткий підсумок виконаного протягом 1984 року.
Програмне забезпечення МВК для здачі технічного комплексу Державній комісії.
Документація програмного забезпечення (8 документів).

Вийшли 2 статті у Збірниках наукових праць Інституту, підготовлених 1983 року.
Тези "Практический подход к построению трансляторов" надіслані до Львова.
Надіслані тези доповіді в Новосибірськ – не їздила.
Виступів 2 – на семінарі у відділі та на конференції в Будинку н/т пропаганди.

----------1985----------

***16 січня** одержала лист з Москви від Селюна М.І. з проханням дати відгук на автореферат кандидатської дисертації "Расширяемый алгоритмический язык АБВ и его реализация", яка виконувалася в ОЦ АН СРСР протягом 8 років. Науковий керівник С.С. Лавров. Захист мав відбутися 21 лютого. Звичайно, я написала позитивний відгук. Селюн М.І. був одним з тих молодих наукових співробітників, хто ознайомився з моєю дисертацією і підписав відгук ОЦ на*

неї у 1973 році, завірений Дородніциним А.О.

***29-31 января**. Республиканская конференция "Надежность и качество программного обеспечения". Учредители: АН СССР, АН УССР, Минвуз УССР, Ордена Ленина ИК им. В.М. Глушкова, Львовский госуниверситет им. И. Франко.*

*Доклад: Н.М. Мищенко **"Практический подход к построению надежных трансляторов"**.*

Аннотация. Суть предлагаемого подхода к построению надежных трансляторов состоит в том, что требуемый транслятор Т для некоторого кс-языка программирования реализуется путем специального расширения заранее созданного и отлаженного универсального программного базиса, являющегося общим в реализации всех КС- языков некоторого класса. Универсальный программный базис – это расширяющаяся система программирования (РСП) ТЕРЕМ, которая содержит программы, общие для трансляторов класса языков, реализуемых с помощью РСП ТЕРЕМ. К таким программам относится прежде всего синтаксическая подсистема, управляемая синтаксическими таблицами, которые строятся системой ТЕРЕМ по БНФ-описанию грамматики реализуемого языка. Имеется также ряд универсальных программ семантической подсистемы. Для получения Транслятора с некоторого языка РСП ТЕРЕМ расширяет неполную семантическую подсистему за счет модулей, созданных по специальной методике.
Многократное повторное использование базиса в виде РСП ТЕРЕМ обеспечивает надежность требуемого транслятора. Объем базиса свыше 3000 операторов языка ПЛ/1 ОС ЕС. В настоящее время РСП ТЕРЕМ используется для реализации языка мультимодульного программирования МАЯК.

На этой же конференции был представлен доклад колеги:

Щеголева Н.Н. "Вопросы надежности организации информационной среды многопроцессорного вычислительного комплекса".

***1 лютого**. На цей час основним замовником МВК ЄС 2701 були ЦАГІ (м. Жуковський) та Воєнно-повітряна інженерна Академія (рос. мовою ВВИА имени Н.Е. Жуковского). Розробники математичного забезпечення МВК їздили в ЦАГІ та Академію читати лекції про математичне забезпечення. Була складена*

ПРОГРАММА

курса лекций по общесистемному математическому обеспечению многопроцессорных вычислительных машин с макроконвейерной организацией вычислений (ЭС 2701)

I Структура общесистемного математического обеспечения – 4 часа

II Операционная система (структура, функции) – 10 часов

III Система программирования (языки, трансляторы, организация работ) – 18 часов

IV Инструментальная система (структура средств, средства моделирования, средства проектированич – 12 часов

V Внутреннее математическое обеспечение – 6 часов

VI Технология решения задач (примеры программ, подготовка данных) – 12 часов. Всего 62 часа.

Лекторский коллектив: Капитонова Ю.В., Летичевский А.А., Гороховский С.С., Годлевский А.Б., Горлач С.П., Берестовая С.Н., Кривой С.Л., Клименко В.П., Мищенко Н.М., Федюрко В.В.

***25 лютого**. Основные темы лекций по системе программирования МАЯК*

1. Входной язык СП МАЯК;

2. Структура СП;

3. Руководство программиста (как пользоваться СП для трансляции);

4. Отладка программ;

5. Подготовка данных для выполнения МАЯК-программ.

***6 березня** – виробнича нарада, де Ю.В. Капітонова озвучила задачі на весь рік:*

I МВК. Разработка ОС и СПП – госкомитетовская тема:

1). Теоретическое обоснование решений, принятых в ОС и СПП – Отчет

2). Состояние научных исследований в нашей стране и зарубежом

3). Чтение леккций в ЦАГИ и Академии им. Н.Е.Жуковского

4). В мае подать материал в книгу, публикуемую Академией им. Н.Е. Жуковского

5). Документация

II. ПРОЕКТ

1). Наша тема: проектирование алгоритмов и программ

2). Отчет по хоз. договору СТРУКТУРА "Технологические средства разработки программ"
III Автоматизация доказательств
IV Публикации: Том произведений В.М. Глушкова, 2 сборника, ж. Кибернетика.

19 квітня. *Вперше виступаю офіційним опонентом на захисті дисертації Алієва Тельмана Мисірогли "Исследование грамматик предшествования и разработка системы построения трансляторов для микроЭВМ", подану на здобуття вченого ступеня канд. физ-мат наук.*

30-31 травня *я була в Москві у Військово-Повітряній Академії ім. М.Є. Жуковського та в м. Жуковський. Читала лекції по МВК.*

Пам'ятаю себе в ролі лектора в дружній атмосфері, багато було питань-відповідей. Пам'ятаю також збори нашого відділу вже в Києві після читання лекцій у замовників макроконвейєра. Юлія Володимирівна була дуже серйозною, виступала і критикувала всіх, а за що, не пам'ятаю, певно, по ділу. Коли дійшла черга до мене, теж прозвучала критика і раптом тим же сердитим голосом було сказано, що мене там похвалили (там, де я читала лекцію майбутнім користувачам МВК). Оцей перехід від критики до похвали без зміни критичного тону тоді видався кумедним. Цікаво, що критикувала Ю.В. всіх, а якщо критика поділена на великий гурт, то кожному її випадає не так уже й багато. Тому й не пам'ятаю нашої спільної провини.

Перед поїздкою до Москви одержала лист від голови Робочої групи з реалізації мов програмування (рос. мовою РГ РЯП) І.В. Поттосіна, у якому він сповістив, що в м. Калінін з 3 по 5 червня відбудеться засідання ЦП РЯиС (цільової підгрупи з реалізації мов програмування і систем). Розміщення в готелі НПО "Центрпрограммсистем" і всі дані, як туди добратися.

Везіння: читання лекцій і засідання підгрупи майже в одному й тому ж місці далеко від Києва. ***3-5 червня****. Калінін. Я відвідала засідання ЦПГ РЯиС без доповіді як член цільової підгрупи. На жаль, не знайшла в інтернеті інформації про жодне з засідань ЦПГ РЯП, що відбулися у 1985 р. У центрі міста була красива церква, мені сказали: єдина діюча. Я вирішила зайти. Це було хвилин за 15 до 6-ої години вечора, коли вона мала закритися. У церкві правив службу священник, кілька літніх жінок співали. За кілька хвилин до 6-ої години священник щез за святими воротами, а півча продовжувала співати. Через лічені хвилини уже два (!) священники буквально вибігли в парадному цивільному одязі і, як личить джентельменам, з модними на той час кейсами і бігцем попрямували до дверей. Несподівана перепона – назустріч у дверях їм трапилися прихожанки, які миттю попадали перед священниками на коліна. Попи дали на ходу поцілувати руки і вибігли з церкви. Мені стало цікаво, куди вони так поспішали. Я вийшла за ними. Вони побігли за церкву, сіли в добротну легкову машину і щезли з виду. Запам'яталося, бо вперше була свідком ігнорування чину.*

5 червня я покинула Калінін. Залишилося дуже приємне враження від міста, а саме, його старої частини. Різьблене дерев'яне оздоблення вікон, дверей, ганків. Привітні люди. Якби я затрималася ще на один день, то могла б побувати на святі дня народженн О.С. Пушкіна. Чула пізніше захоплені відгуки про це свято. Не згадала завчасно.

*Подаю технічне завдання на дослідно-конструкторську роботу "****Единая система ЭВМ****.* ***Многопроцессорный вычислительный комплекс с макроконвейерной обработкой данных ЕС 1766****" – найпотужніший комплекс МВК з макроконвейєрною обробкою даних, що став останнім у нашій тодішній історії з ЕОМ ЄС.*

ЕС ЭВМ

МВК с макроконвейерной обработкой данных ЕС 1766

Опытно-конструкторская работа

Техническое задание

ЕС 1766 на базе ЕС 1066 и спецпроцессоров ЕС 2701, ЕС 2680 с переменной конфигурацией разрабатывается на основании Решения Комиссии Президиума Совмина СССР от 16.08.84 г. №285, приказа руководителя организации п/я М-5804 от 30.08.84 №520.

31 травня 1985 року *було _____затверджене Технічне завдання (ТЗ) після його узгодження 13-мая організаціями, від імені яких свої підписи поставили, зокрема, (російською):*

В.А. Мясников (ГКНТ), К.Н. Трохимов (в/ч 52686), Н.В. Горшков (п/я М-5804), Ю.Т. Семихов (п/я В-8325), В.В. Пржиялковский (зам.ген. конструктора ЕС ЭВМ), В.С. Михалевич (директор ИК им. В.М. Глушкова АН УССР) и другие. Затвердив ТЗ керівник п/я М-5804 П.С. Плешаков.

Наводжу пункти ТЗ, до яких програмісти мали безпосереднє відношення.

4.4. Принципы работы ЕС 1766

4.4.4. Технические и программные средства ЕС 1766 должны обеспечивать:

– режим параллельной обработки большого объема данных с высокой производительностью

– мультипрограммную работу

– диалоговую обработку заданий

– работу в реальном масштабе времени

– автоматическую реконфигурацию ЕС 2701 и продолжение вычислений при отказах отдельных процессоров, входящих в его состав.

4.5. Программное обеспечение

4.5.1. Программное обеспечение ЕС 1766 должно обеспечивать:

– выполнение функций в соответствии с п. 4.4.4. настоящего ТЗ

– возможность подготовки и отладки программ для ЕС 1766 с использованием языка программирования высокого уровня

– совместное функционирование с ЕС 1066, ЕС 2701, ЕС 2680 в составе ЕС 1766.

4.5.2. В состав ПО ЕС 1766 должны входить следующие компоненты:

– система ПО ЭВМ ЕС 1068, ЕС 2701, ЕС 2680

– распределенная операционная система ЕС 1766

– система параллельного программирования ЕС 1766

– инструментальная система проектирования и моделирования програмных систем

– комплекс программ технического обслуживания ЕС 1766

– пакеты прикладных программ для задач вычислительной математики и оптимизации.

4.5.6. Базовым языком параллельного программирования является язык МАЯК, обеспечивающий параллельную обработку и содержащий широкий набор средств параллельного программирования, ориентированных как на прикладных, так и на системных программистов. В том числе обеспечивающий возможность разработки программ, допускающих динамическое распараллеливание вычислительных процессов.

4.5.7. СП ЕС 1766 должна допускать в программах использование модулей, написанных на стандартных языках программрования ФОРТРАН (Гост 23056-78), КОБОЛ (Гост 22558-78) и ПЛ/1.

4.5.8. СП должна допускать раздельную компиляцию компонент программ и их независимую отладку.

4.5.9. Система проектирования и моделирования программных систем должна обеспечивать инструментальную поддержку технологии разработки программ и программных систем, ориентированных на макроконвейерную организацию вычислений.

6.1. Стадии и этапы разработки. Разработка ЕС 1766 должна проводиться по этапам:

1. Разработка ТП (технического проекта) ЕС 1766; IV кв. 1985 г.

2. Разработка конструкторской документации II кв. 1986 г.

3. Изготовление и наладка опытного образца ЕС 1766; III кв. 1986 г.

4.Проведение предварительных испытаний и предъявление на госиспытания; IV кв.1986

6.2. Исполнители и изготовители

6.2.1. Головной исполнитель ОКР "Разработка высокопроизводительного комплекса ЕС 1766 (на базе ЭВМ ЕС 1066 и спецпроцессоров ЕС 2701, ЕС 2680) с переменной конфигурацией, производящий от 30 до 100 млн. эквивалентных ЕС ЭВМ команд в секунду" – предприятие п/я М-5769.

Ответственный исполнитель ИК АН УССР.

Исполнитель СКБ ММС. Изготовитель ЕС 1766 – предприятие п/я А-7182.

Межведомственную комиссию для испытаний назначает п/я М-5804.

Руководитель организации П.С. Плешаков, заместитель В. Н Горшков.

***2 жовтня**. Виступ з колегами на семінарі Наукової ради ІК АН УРСР з проблеми "Кібернетика".*

Секція 1. Семінар 1.1. Теорія автоматів та її застосування.

Н.М. Мищенко, В.В. Федюрко, Е.А. Дудко "Расширение ПЛ/1 средствами взаимодействия МВК с внешней памятью".

Т.А. Валькевич "Словари мультимодульных программ".

Обидві доповіді 11 червня 1986 р. були підписані до друку і вийшли в збірнику

Проектирование многопроцессорных вычислительных систем: Сб. науч. тр. – Киев: Ин-т кибернетики им. В.М. Глушкова АН УССР, 1986.

В статье Н.М. Мищенко, В.В. Федюрко, Е.А. Дудко рассмотрен языковой аспект средств

взаимодействия компонент мультимодульных программ с внейшней памятью МВК и их реализация в виде расширения языка ПЛ-1 с помощью РСП ТЕРЕМ. Реализация такого расширения позволяет автоматизировать процесс разработки общесистемного математического обеспечения МВК и демонстрирует методику применения системы ТЕРЕМ для построения языковых процессоров, ориентированных на класс языков, описываемых наборами схем предложений (образцов).

В статье Т.А. Валькевич рассматриваются словари мультимодульных программ, реализованные в составе СПП МАЯК и предназначенные для сбора, хранения и использования информации об именованных объектах входных программ. Словари реализованы как абстрактные структуры данных, определяемые с помощью операторов обращения к ним. Выделены универсальные операторы, которые можно использовать при реализации других языков программирования.

Листопад 1985 року. Відзначаємо ювілей Ю.В. Капітонової – 50-ліття (нар 21 листопада).
На фото: Олексій Кучеренко, Сергій Коляда, Володимир Чуйкевич, Сергій Горлач, ?, Сергій Морозов, стоїть Надія Міщенко, невидима за нею Наталя Щоголева, далі – Ольга Феліжанко, Марина Байда, Тетяна Поліщук, Людмила Черкасова, Наталя Соболєва, Тетяна Шевелюк.
***26-27 листопада** здали тему – держкомітетівську науково-дослідну розробку.*
Название темы: "Разработать и сдать в опытную эксплуатацию ОС и СПП с языком высокого уровня для динамического распаралелливания в процессе выполнения программ на МВК с макроконвейерной организацией вычислений" (заключ. отчет).
Шифр темы: РН.80.00.45. Шифр проблемы 0.80.14.
Наименование отчета "Операционная система и система параллельного программирования для вычислительных комплексов с макроконвейерной организацией вычислений".
Раздел 5. Система параллельного программирования (біля 12 стор.).
Члены комиссии:
директор ВЦ АН СССР академик Дородницын А..А. – председатель комиссии,
начальник ВЦ ЦАГИ, доктор техн. наук Смирнов А.Д. – заместитель председателя,
зам. директора Института кибернетики им. В.М. Глушкова АН УССР, член- корр. АН УССР Сергиенко И.В., зам. председателя,
старший научный сотрудник ЦАГИ, кандидат физ.-мат. наук Бушуева И.М.,
старший научный сотрудник ВЦ АН СССР Чарахчьян А.А.
Первый этап экспериментальной реализации проведен для таких прикладных задач.

1. *Модуль расчета переноса-диффузии для задачи моделирования климата Мирового океана*
2. *Задача гиперзвукового обтекания твердого тела*
3. *Решение системы линейных алгебраических уравнений методом Холецкого*
4. *Задача расчета плоского деформированного состояния*
5. *Задача оптимального планирования производства и распределения труб*
6. *Решение квазилинейного эллиптического уравнения*
7. *Задача линейного программирования*
8. *Задача булевого линейного программирования*
9. *Задача расчета обтекания цилиндра*
10. *Решение задачи Дирихле для эллиптического уравнения*
11. *Решение параболического уравнения*
12. *Расчет трансзвукового течения*
13. *Решение квадратичной задачи о назначениях*
14. *Решение линейных интегральных уравнений.*

Приведенный перечень включает в себя задачи математической физики, линейной алгебры и оптимизационные задачи. Исходной информацией при реализации являлись Фортран-программа (для задач 1 и 2) или математическая постановка для остальных задач.
Автономная отладка проведена для задач: 1-4, 9, 10.

До кінця року: звіт та його оформлення, обговорення статей з Т.Валькевич, С.Кривим, О. Дудко,
з В. Федюрко про приклади для статтей.

----------1986----------

На початку року у відділах ТЦА та РВМ почали виконуватися кілька тем, затверджених ГКНТ (рос. абревіатура для Державного комітету по науці і техніці). Наведу ті з них, у виконанні яких я з колегами брала участь.

***ПРОБЛЕМА**: Создать и освоить в производстве вычислительные комплексы общего назначения, управляющие и проблемно-ориентированные вычислительные комплексы, периферийное оборудование и программные средства для них.*

ТЕМА: Разработать и создать экспериментальные образцы процессоров логического вывода и математического обеспечения для вычислительной системы с макроконвейерной обработкой данных.

Шифр: 0.80.01.24.01. И -- С.Г. 100.02 -- ГКНТ СССР № 555 от 30/X-85 – I кв. 8685–IV кв. 90

ТЕМА: Создать и ввести в эксплуатацию экспериментальную систему для автоматизированного проектированного ЭВМ новых поколений на СБИС с уровнем интеграции 10 млн на кристалл.

А2. Техническое задание на САПР ЭВМ на СБИС

1. Определение состава технологических звеньев проектирования в САПР ЭВМ на СБИС

2. Разработать техническое задание на САПР на СБИС3. Выбор и обоснование рационального варианта САПР

4. Разработка входного языка и механизмов функционирования основных компонент САПР.

Шифр: 0.80.01.24.08. А -- С.Г.100.03 -- ГКНТ СССР № 555 от 30/X-85 -- I кв. 8685–IV кв.89

ТЕМА: Провести научно-исследовательские работы по вычислительным системам 5-го поколения и создать экспериментальные образцы электронных приборов, интеллектуальных процессоров и систем обработки информации нового поколения

Шифр: 0.80.01.24 -- С.Г.100.05 -- ГКНТ СССР № 555 от 30/X-85 -- I кв. 8685–IV кв.90

ТЕМА: Создать и освоить в производстве многопроцессорную ЭВМ ЕС 1710 с макроконвейерной обработкой данных с производительностью 500 млн команд/сек, эквивалентных ЕС ЭВМ, предназначенную для решения проблеммных задач большой размерности в режиме коллективного диалога с пользователем

Шифр: 0.80.01.21 -- С.Г.100.06 -- ГКНТ СССР № 555 от 30/X-85 -- I кв. 8685–IV кв.90

***ПРОБЛЕМА**: Создать новые и развить действующие системы автоматизированного проектирования (САПР) и автоматизированные системы научных исследований (АСНИ) в народном хозяйстве (автоматизация исследований)*

ТЕМА: Создать и ввести в опытную эксплуатацию в Институте кибернетики имени В.М. Глушкова АН УССР САПР высокопроизводительных ЭВМ с аппаратной реализацией системного программного обеспечения на основе перспективной элементной базы.

Сформулировать и обосновать требования к САПР.

1. Подготовить аналитический обзор известных подходов к разработке и использованию САПР

ЭВМ.

2. Сформулировать основные задачи проектирования высокопроизводительных ЭВМ и разработать методы их решения, ориентированные на автоматизацию проектирования.

3. Сформулировать задачи создания, использования и развития базовой САПР для реализации системной технологии проектирования ЭВМ

4. Сформулировать технические требования к САПР.

Шифр: 0.80.03.01.34 А -- С.Г.100.04 -- ГКНТ СССР № 573 от 10/XI-85 -- I кв. 8685–IV кв.90

Цього року почалося остаточне укомплектування макроконвейєра високопродуктивною обчислювальною технікою. Було виготовлено процесор ЄС 1766.

Наведится окумент, що засвідчує завершення розробляння Макроконвейєра.

ЕС ЭВМ

МВК с макроконвейерной обработкой данных ЕС 1766

Опытно-конструкторская работа

Техническое задание

ЕС 1766 на базе ЕС 1066 и спецпроцессоров ЕС 2701, ЕС 2680 с переменной конфигурацией разрабатывается на основании Решения Комиссии Президиума Совмина СССР от 16.08.84г. №285, приказа руководителя организации п/я М-5804 от 30.08.84 №520.

4.4. Принципы работы

4.4.4. Технические и программные средства ЕС 1766 должны обеспечивать:
–режим параллельной обработки большого объема данных с высокой производительностью
–мультипрограммную работу
–диалоговую обработку заданий
–работу в реальном масштабе времени
–автоматическую реконфигурацию ЕС 2701 и продолжение вычислений при отказах отдельных процессоров, входящих в его состав.
4.5. Программное обеспечение
4.5.1. Программное обеспечение ЕС 1766 должно обеспечивать:
–выполнение функций в соответствии с п. 4.4.4. настоящего ТЗ
–возможность подготовки и отладки программ для ЕС 1766 с использованием языка программирования высокого уровня
49
– совместное функционирование с ЕС 1066, ЕС 2701, ЕС 2680 в составе ЕС 1766.
4.5.2. В состав ПО ЕС 1766 должны входить следующие компоненты:
– система ПО ЭВМ ЕС 1068, ЕС 2701, ЕС 2680
– распределенная операционная система ЕС 1766
– система параллельного программирования ЕС 1766
– инструментальная система проектирования и моделирования программных систем
– комплекс программ технического обслуживания ЕС 1766
– пакеты прикладных программ для задач вычислительной математики и оптимизации.
4.5.6. Базовым языком параллельного программирования является язык МАЯК, обеспечивающий параллельную обработку, в том числе возможность разработки программ, допускающих динамическое распараллеливание вычислительных процессов, и содержащий широкий набор средств параллельного программирования, ориентированных как на прикладных, так и на системных программистов.
4.5.7. СП ЕС 1766 должна допускать использование в программах модулей, написанных на стандартных языках программрования ФОРТРАН (Гост 23056-78), КОБОЛ (Гост 22558-78) и ПЛ/1
4.5.9. Инструментальные средства
6.1. Стадии и этапы разработки
Разработка ЕС 1766 должна проводиться по следующим этапам:
1. Разработка ТП ЕС 1766; IV кв. 1985 г.
2. Разработка конструкторской документации II кв. 1986 г.
3. Изготовление и наладка опытного образца ЕС 1766; III кв. 1986 г.
4.Проведение предварительных испытаний и предъявление на госиспытания; IV кв. 1986 г.
6.2. Исполнители и изготовители.
6.2.1. Головной исполнитель ОКР "Разработка высокопроизводительного комплекса ЕС 1766 (на базе ЭВМ ЕС 1066 и спецпроцессоров ЕС 2701, ЕС 2680) с переменной конфигурацией, производящий от 30 до 100 млн эквивалентных ЕС ЭВМ команд в секунду" – предприятие п/я М-5769
Ответственный исполнитель ИК АН УССР
Исполнитель СКБ ММС. Изготовитель ЕС 1766 – предприятие п/я А-7182
Межведомственную комиссию для испытаний назначает п/я М-5804
Рук. организации П.С. Плешаков, заместитель Н.В. Горшков.
Випробування нового обчислювального комплексу відбудеться в кінці року, а тим часом займаємося поточними справами.
***18 січня** підписана до друку монография:*
"Системное математическое обеспечение многопроцессорного вычислительного комплекса ЕС". 390 стр. Издана в типографии ВВИА имени проф. Н.Е. Жуковского.
Авторів більше 30-ти, їхні прізвища наводяться в алфавітному порядку. Виняток – для директора Михалевича В.С: він очолює список, а закінчує його К.Л.Ющенко.
Глави, в написанні яких я брала участь (біля 50 сторінок):
Глава 4. АВТОМАТИЗАЦИЯ ПРОГРАММИРОВАНИЯ (сс.132-153).
Глава 6. КОМПЛЕКС ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ ПРОЕКТИРОВАНИЯ И МОДЕЛИРОВАНИЯ.

Раздел 6.6. Система проектирования языковых процессоров (сс. 358-386).

***15 лютого**. Сесія у відділі. Суботник. Підведення підсумків по СПП МАЯК.*

ВИКОНАНО:

1. Создан технический проект СПП МАЯК. Основные решения проекта обоснованы с учетом:

– особенностей языка МАЯК;

– решений, принятых в ОС;

– конфигурации МВК;

– наличных инструментальных средств и имеющегося коллектива исполнителей.

2. Проект реализован, в том числе:

– разработаны концепции и алгоритм декомпозиции ММ-программы на простые модули (программа Декомпозитор, авторы Мищенко, Дудко);

– разработаны концепции, алгоритм и выполнена реализация словарей ММ-программ в виде автономной системы процедур, готовой к включению в трансляторы САА (Мищенко, Валькевич);

3. Проведена опытная эксплуатация Декомпозитора (первый этап СПП Дудко Е.А.);

4. В ходе реализации и опытной эксплуатации сформирован синтаксис входного языка. Он отличается от опубликованного языка МАЯК не только наложенными ограничениями наязык МАЯК и уточнениями в соответствии с конфигурацией МВК, но и с изменениями некоторых понятий, представленных в публикации по языку. (Дудко, Валькевич, Мищенко)

5. Описан синтаксис системного уровня входного языка в метаязыке РСП ТЕРЕМ и трансляционная семантика (Мищенко);

6. Публикации по всем частям СПП: ММП-декомпозитор, УПР-, АПР-трансляторы;

7. Сдача комиссии: программирование и литература (Мищенко), отчет, документы, примеры (Дудко).

ВИКОНАТИ:

1. Описать СПП МАЯК в окончательном виде: функции и структура. Единственная публикацияов в 1982 году отражает поиск решений, а не сами решения, принятые и уже реализованные (с авторами языка Капитоновой Ю.В., Летичевским А.А., С.С. Гороховским)

2. Описание входного языка для пользователей (Гороховский С.С.)

3. Развитие ММП-декомпозитора: с Валькевич – внедрение словаря, снятие некоторых ограничений, с Фелижанко – директивы запуска системы, с Федюрко – упорядочение порций программы, со Щеголевой – генерация модуля ВЗМ

4. Отладка синтаксиса системного уровня входного языка (Дудко Е.А.)

5. Обработка данных, полученных при опытной эксплуатации СПП (Дудко Е.А.)

6. Продолжить опытную эксплуатацию (Дудко Е.А.)

7. Документация для пользователей (Мищенко Н.М., Дудко Е.А.)

***20 квітня** переатестація, для якої подано звіт наступного змісту:*

О Т Ч Е Т

старшего научного сотрудника Мищенко Н.М., отдел № 105981 – 1986)

1. Научно-производственная работа

1.1. Проведены исследования в области расширяющихся языков и систем программирования, а также в области проектирования средств автоматизации программирования для многопроцессорных вычислительных систем. Разработала алгоритм и реализовала инструментальную расширяющуюся систему программирования ТЕРЕМ, предназначенную для разработки языковых процессоров, выполняющих контекстно-свободные атрибутные переводы. Развивала и сопровождала эту систему при построении транслирующих систем.

1.2. Участвовала в разработке структуры системы параллельного программирования для языка МАЯК в рамках общесистемного математического обеспечения МВК с макроконвейерной организацией вычислений. Разработала алгоритм и программу Декомпозитора мультимодульных программ.

Сопровождала инструментальную РСП ТЕРЕМ в процессе разработки сотрудниками отдела трех трансляторов, входящих в СПП МАЯК.

2. Должностная и методическая работа

Выполняла работы по темам ГКНТ, принимала участие в выполнении хоздоговоров. Участвовала в разработке технических заданий, написании научно-технических отчетов, программной документации. Читала лекции по МВК пользователям из ИК АН УССР, ВВИА

им. Н.Е Жуковского (г. Москва) и ЦАГИ (г. Жуковский). Руководила практикой и дипломными работами студентов, рецензировала статьи и авторефераты диссертаций, выступала оппонентом на защите кандидатской диссертации.

3. Публикации и конференции.

Опубликовала 9 работ, из них 2 в центральных издательствах.
Выступала с докладами на двух Всесоюзных конференциях, 7 республиканских семинарах и на Рабочей группе по реализации языков программирования (2). Участвовала в написании монографии, посвященной общесистемному математическому обеспечению МВК.

4. Общественная работа. На протяжении последних 4 лет была профгруппоргом трудового коллектива отделов №100 и 105.

***3-5 червня в Кишиневі** в Університеті відбулася нарада-семінар цільової підгрупи з розширних мов програмування. Лист про цей захід я одержала зимою і поїхала б, якби не катастрофа в Чорнобилі. У цей час я відвозила дітей до родички у Львів.*
Чи працює Цільова підгрупа з розширних мов програмування в Росії після розпаду СРСР, мені невідомо. ВІдомо, що немає вже багатьох, хто цю роботу підтримував.

***5 серпня**. Цього року завершувалося комплектування високопродуктивного обчислювального комплексу ЄС 1766. Технічне завдання на його виготовлення наведено у спогадах за 1985 рік. Виготовлення і налагодження дослідного зразка ЄС 1766 мало відбутися у III кварталі цього року, а попередні і державні випробування – у IV кварталі. Зазделегідь була створена Міжвідомча комісія для приймання технічного проекту обчислювального комплексу ЄС 1766, яка перевірила стан готовності ПО (ПО – російською программное обеспечение) і зробила ряд зауважень. Комісія знайшла у виконанні пп. 5, 6, 8, 9 ТЗ такі недоліки (у дужках – запланований термін виправлення недоліків):*

5. Не описаны а) состав и формы дистрибутивных материалов ПО ЕС 1766; б) порядок развертывания этих материалов на конкретных образцах ЕС 1766 (ноябрь 1986).
Не описан состав эксплуатационных документов ПО (июль 1986).
Недостаточно конкретно указаны особенности языков программирования, входящих в состав ПО, в частности, в связи с необходимым удовлетворением ТЗ и реализации стандартизированных версий языков ФОРТРАН и КОБОЛ (август 1986).

6. Не отработаны вопросы функционирования ПО под управлением новых версий ОС ЕС (в частности, ОС 7) (ноябрь 1986).

8. Транслятор с ФОРТРАН-4 на АПР не в полной мере реализует язык ФОРТРАН (ноябрь 1986).

9. Не представлена концепция тестирования программного обеспечения ЕС 1766 (ноябрь 1986).

***8-13 сентября**. 7**-а Всесоюзна школа-семінар "Параллельное программирование и высокопроизводительные системы".** Учредители: ГКНТ СССР, Президиум АН СССР, Институт кибернетики им. В.М. Глушкова АН УССР, Симферопольский госуниверситет.*

Тези доповідей розділені за темами:

Общие вопросы структурного параллельного программирования.

Многоуровневое структурное проектирование параллельных программ.

Математическое обеспечение многопроцессорных вычислительных комплексов.

Прикладное параллельное программирование.

доповідь віднесена до першої теми.

Мищенко Н.М. Некоторые особенности реализации языка МАЯК // Параллельное программирование и высокопроизводительные системы. Тез. докл. 7 Всесоюз. школы-семинара. Киев: Ин-т кибернетики им. В.М. Глушкова АН УССР, 1986. – С.121-122

Короткий зміст доповіді. Основным языком программирования для многопроцессорного вычислительного комплекса (МВК) с макроконвейерной организацией вычислений является согласованное семейство языков (язык МАЯК), реализованное в системе программирования (СП) МАЯК.

Кроме функции – трансляции СПП МАЯК выполняет ряд других функций.

1. Основное отличие языка МАЯК от других ЯП определяется большим набором структур управления, ориентированных на параллельное программирование для МВК с общей и распределенной памятью и обеспечивающих макроконвейерную обработку данных.

52

2. Поддержка ОС в языке МАЯК состоит в том, что в ОС реализованы базовые структуры управления асинхронными вычислениями, такие как передача сообщений, синхронизация и защита общих данных, не имеющие аналогов во внутренних языках процессоров МВК.

3. Ввиду неоднородности МВК входные МАЯК-программы состоят из модулей разных типов в соответствии с типами процессоров, на которых они должны выполняться. Для написания модулей используются специальные подмножества языка МАЯК. Это определяет СП как систему трансляторов и добавляет в СП еще одну функцию – предварительное расчленение (декомпозицию) ММ-программы на отдельные модули.

4. Определение областей локализации переменных и построение на его основе областей видимости объектов в МАЯК-программе накладывает жесткие ограничения на порядок трансляции модулей: сначала следует транслировать самый внешний модуль, затем – подчиненные внешнему и т. д.

С учетом вышесказанного трансляция ММ-программ в СПП МАЯК осуществляется в 3 этапа (см. Блок-схему СПП МАЯК).

1985 рік. Блок-схема трансляції ММП-програм на заключному етапі побудови СП МВК

На первом, машинно-независимом этапе выполняется декомпозиция ММ-программ на отдельные модули типа АПР, УПР или ППР.

Второй этап – это раздельная и независимая трансляция АПР-, УПР- и ППР-модулей в языки системного уровня АПР-0, УПР-0 и ППР-0, соответственно, а также генерация в языке УПР-0 управляющих программ ОС.

53

Третий этап – машинно-зависимый и состоит в трансляции модулей в языках АПР-0, УПР-0 и ППР-0 в языки соответствующих процессоров.

Экспериментальная версия СП МАЯК функционирует в ОС ЕС ЭВМ.

***2 листопада.** 4 рівні представлення системного МО перед комісією.*

Опытный образец (декабрь 1986 года). Минимум из того, что уже работало – АПР-0. Пользовательский уровень (отражаем в документации). Методика реализации МАЯКа с помощью МАЯК-0. УПР-0 – СМ-овский вариант

Дополнительные средства, расширяющие возможности штатного СМО, (Декомпозиция, АПР-0, МАЯК-0). Модернизация существующего МО ОС 7. Реализация СП МАЯК.

18-21 листопада. II Всесоюзна конференція "Технология программирования".

Учредители:

Госкомитет СССР по науке и технике, Госкомитет СССР по выч. технике и информатике, АН СССР, АН УССР, Научно-техническая комиссия ГКНТ по технологии программирования, ИК им. В.М. Глушкова АН УССР. Киев.

Підрозділи конференції:

Общие вопросы методологии создания программных средств

Методы проектирования программных средств

Технология проектирования в системе образования

Правовые, экономические и нормативно-методические вопросы технологии программирования.

Наша доповідь віднесена до першого розділу:

Е.А. Дудко, Н.М. Мищенко "Декомпозиция мультимодульных программ в языке МАЯК".

Коротко зміст доповіді. Подготовка задач к решению на неоднородном многопроцессорном вычислительном комплексе (МВК) с макроконвейерной организацией вычислений состоит из двух технологических этапов. На первом, неавтоматизированом этапе анализируются вычислительные методы и выбирается оптимальная организация параллельных вычислений. Результат этапа – мультимодульная программа решения задачи в языке параллельного программирования МАЯК.

Второй этап выполняется автоматически системой программирования МАЯК (СП МАЯК), которая имеет ряд особенностей. Так, она не только транслирует мультимодульную программу в языки процессоров МВК, но и генерирует специальный уровень операционной

системы для данной программы. Кроме того, в СП МАЯК трансляции и генерации предшествует работа Декомпозитора – языкового процессора, который выполняет декомпозицию мультимодульных программ на простые модули, которые затем транслируются ввиду неоднородности МВК разными трансляторами СП МАЯК.

Далі наведені вище етапи деталізуються.

На пленарному засіданні виступав . Його думка щодо технології програмування: Законченная технология программирования должна:

– охватывать весь жизненный цикл программного продукта;

– способствовать применению методологии, повышающей уровень достоверности, надежности и доказательности программирования;

– опираться на современные технические средства в виде автоматизированных рабочих мест, объединяемых в локальную сеть;

– обеспечивать организационную управляемость и контролируемость производственных процессов;

– обеспечивать устойчивость программного продукта по отношению к смене технических средств;

– обеспечивать развиваемость программного продукта в связи с изменением условий функционирования целевой системы, использующей этот продукт.

Полезно различать в перспективе до 2000 года три поколения интегральной промышленной технологии программирования.

Первое поколение (организационное программирование) 1975-1985 гг. ФОРТРАН, КОБОЛ, ПЛ-1, ассемблер.

54

Третье поколение (доказательное программирование) 1995-2005 гг.

Так я вперше познайомилася з терміном "сборочное программирование".

***27 листопада**. Збори. Найперше питання – наказ директора: ЄС 1766 – здавати Комісії. Процесор ЄС1766 у моїх щоденниках більше не згадується, певно, здали успішно.*

***Грудень.** Третя Всесоюзна конференція "Автоматизация производства систем программирования" Таллін, Інститут кібернетики АН ЭРСР, ул. Академія тее, 21. Представила тези доповіді: **"Реализация контекстно-зависимых переводов в системе ТЕРЕМ"** на трьох сторінках. Одержала відмову за 11.10.1986 року за підписом Я.Э. Пеньяма. Основна ідея доповіді, що не відбулася, полягала в наступному.*

Общим принципом построения языковых процессоров с помощью РСП ТЕРЕМ является использование ее программного базиса в качестве ядра нового процессора и расширение этого ядра синтаксическими таблицами реализуемого и базисного языков и семантическими программами до получения полного процессора. Синтаксические таблицы строятся Конструктором системы ТЕРЕМ автоматически по описанию синтаксиса в метаязыке системы, который является модификацией Бэкусово-Науровских форм. Семантические программы, отражающие специфику функции перевода с реализуемого языка, разрабатываются пользователями. При этом система ТЕРЕМ предоставляет семантическим программам все необходимое для них информационное окружение, а также управляет выполнением семантических программ. Связь программных средств, разрабатываемых пользователем, с программным базисом системы ТЕРЕМ реализована в специальной процедуре, единственной в базисе, подлежащей изменению при использовании системы.

***Грудень.** Напередодні 1987 Нового року Ю.В. Капітонова поширила серед співробітників анкету з різними питаннями стосовно роботи і відпочинку. Питання не збереглися, проте збереглися мої відповіді, з яких легко здогадатися, якими були питання. Наводжу свої відповіді, серед яких є дані про виконану роботу протягом року.*

А Н К Е Т А

1. Міщенко Надія Михайлівна

2. Ст.н.с., 300 рублей в месяц

3. Автоматизация программирования

4. На протяжении года выполнено:

4.1. Подытожена работа по исследованию принципов реализации языка МАЯК, выполненная

на протяжении последнизх 2-3 лет вместе с авторами языка в виде описания проекта соответствующей системы параллельного программирования.
4.2. Исследованы принципы реализации средств синтаксической отладки в РСП ТЕРЕМ, отличающихся от аналогичных средств в других транслирующих системах. Результат оформлен в виде текста и включен в документацию
4.3. Проведено развитие и модернизация РСП ТЕРЕМ
5. Публикации: две публикации тезисов на конференциях, статья в сборнике и два подраздела в книге. Подготовлено к печати три статьи.
6. Принимала участие в двух конференциях (в Киеве и в Бердянске)
7. Картотека: языки программирования, РСП, трансляторы, СПТ; разделы: труды В.М. Глушкова и сотр., А.П. Ершова, отчеты, зарубежные публикации
8. Профгруппорг
9. Проект ВЕГА, распространение персональных компьютеров, макроконвейерные вычисления, монография Поттосина и Касьянова.
10. Путешествовать з детьми. Подшивать листинги успешных результатов на работе.
11. Мои замечания и предложения касаются меня, поэтому я их не привожу.
12. Мои планы – в календарных планах и в соцобъязательствах отдела. Хотелось бы исследовать сборочное программирование. Описать то, что есть у нас и что еще нужно нам сделать в этом направлении. И, конечно, узнать, что уже имеется по этой теме в других коллективах.
55
13. Не представляю скорого развития в области моих научных интересов без надежной техники. Развитие должно идти по пути: проще, надежнее, конструктивнее, простые ЯП, надежные трансляторы, модульное программирование.

----------1987----------

На початку січня був складений графік робіт з математичного забезпечення Макроконвейєра з
вказівками термінів їх виконання.

Системы программирования:
Модернизация транслятора УПР-0 в части перевода управляющих конструкций. Кривой С.Л. 16.03

Модернизация АПР-0 транслятора в части расширения состава встроенных функций. Берестовая С.Н. 2.03.

Коррекция транслятора ФОРТРАН-АПР по результатам предварительных испытаний в части
уточнения входного языка. Стаценко, Кучеренко 9.03.

Отладка транслятора МАЯК-0. Мищенко Н.М., Годлевский А.Б. 23.03.
Коррекция программной документации на МЛ. Мищенко Н.М., Берестовая С.Н. 16.03.

Инструментальный комплекс:
Адаптация РСП ТЕРЕМ на ЕС 1066. Мищенко Н.М. 16.03.
Коррекция динамического распараллеливания. Крат С.П., 9.03.

***11 лютого**. Виступ на семінарі Наукової ради АН УРСР з проблеми "Кибернетика". Секція 1. Семінар 1.1. Теорія автоматів та її застосування.*

Доповіді: Н.М. Мищенко "Синтаксическая отладка в СПТ с нисходящим анализом, допускающим ограниченные возвраты."

В основе рассматриваемых средств синтаксической отладки лежит пропуск фраз с ошибками до некоторых специальных символов и продолжение анализа входного текста.

Будучи составной частью РСП ТЕРЕМ, выступающей в роли системы построения трансляторов, средства синтаксической отладки опираются не на лексику реализуемого языка, а на общие правила описания грамматик, что позволяет включать их без изменения в разные языковые процессоры, разрабатываемые средствами системы.

За матеріалом доповіді надрукована стаття у збірнику ІК АН УРСР, підписана до друку 10 червня 1988 року.

Мищенко Н.М. Синтаксическая отладка в СПТ с нисходящим анализом, допускающим

ограниченные возвраты // Методы и средства проектирования дискретных систем: Сб.науч. тр. – Киев: Ин-т кибернетики именм В.М. Глушкова АН УССР, 1988. – С. 49-57.
На цьому ж семінарі відбулася доповідь О.А. Дудко "Расширение функций декомпозитора системы программирования МАЯК" (текст не зберігся).

15-17 вересня. *І Всесоюзна конференція "Проблемы создания супер-ЭВМ, супер-систем и эффективность их применения". К 85-летию академика С.А. ЛЕБЕДЕВА. Учредители: АН СССР, Отделение информатики, вычислительной техники и автоматизации АН СССР, Институт проблем кибернетики АН БССР, Минский радиотехнический институт*

Доповіді розподілені по 5 секціях:

Секция 1. Архитектура супер-ЭВМ;

Секция 2. Системное программное обеспечение и автоматизация проектирования;

Секция 3. Прикладное программное обеспечение супер-ЭВМ;

Секция 4. Конструкция и элементная база супер-ЭВМ;

Секция 5. Надежность, контроль и диагностика супер-ЭВМ.

Дві доповіді з моєю участю були проголошені на Секції 2:

Коротко зміст доповіді. Назначение ОС и СП и соотношение между ними как взаимосвязанными частями общесистемного математического обеспечения определяется классом реализуемых в этих системах функций, среди которых выделим распределенные,

реализуемые как в ОС, так и в СП или в обеих системах сразу. Классификация распределенных функций основана на понятии времени связывания и включает в себя функции с фиксированным, управляемым и разделяемым временем связывания. Для изучения проблемы взаимодействия ОС и СП наиболее интересны функции с разделяемым временем связывания. Их характерная черта в том, что они реализуются и в ОС, и в СП.
Для реализации распределенных функций решающее значение имеют совместное проектирование ОС и СП, использование информационной среды, содержащей данные об отдельных модулях ММ-программы, обеспечение асинхронной работы модулей ММ-программы и взаимодействие модулей посредством языка сообщений.
Н.М. Мищенко "Применение РСП ТЕРЕМ для реализации языка параллельного программирования МАЯК".

Коротко зміст доповіді. Язык параллельного программирования МАЯК представляет собой согласованное семейство языков, предназначенных для решения задач на многопроцессорных вычислительных комплексах (МВК) с макроконвейерной организацией вычислений. Рассматриваемые МВК имеют три типа процессоров с разными внутренними языками: управляющие (УПР), арифметические (АПР) и периферийный (ППР). Соответственно, язык МАЯК включает языки МАЯК-УПР, МАЯК-АПР, МАЯК-ППР.
Для реализации трансляторов с приведенных выше языков была использована расширяющаяся система программирования ТЕРЕМ, в наличии которой имеется широкий спектр средств расширения от макротехники до метасредств систем построения трансляторов. С ее помощью были реализованы:

1. Декомпозитор, выполняющий расчленение мультимодульных программ на отдельные программные модули типа УПР, АПР, ППР. Тем самим декомпозитор обеспечивает дальнейшую раздельную трансляцию полученных модулей на языки соответствующих процессоров.

2. Транслятор с языка МАЯК-УПР во внутренний язык управляющего процессора.

3. Язык МАЯК-АПР реализован как расширение языка МАЯК-АПР0.

4. Транслятор с языка МАЯК-ППР на язык ПЛ-1 ОС ЕС.

Опыт применения РСП ТЕРЕМ показывает следующее:

– включение многократно проверенного программного базиса РСП ТЕРЕМ в качестве ядра реализуемого транслятора увеличивает надежность последнего;

– табличная связь семантических программ с элементами синтаксиса обеспечивает оперативность внесения изменений в языковые процессоры при их разработке и отладке;

– модульная структура программного базиса позволяет исключать процедуры расширения после отладки транслятора, если входной язык не изменяется в процессе использования.

Грудень. *Вийшла друком підсумкова стаття в журналі "Управляющие системы и машины": Ю.В. Капитонова, А.А. Летичевский, Н.М. Мищенко* ***"Расширяющаяся система программирования ТЕРЕМ – инструмент для разработки языковых процессоров на ЕС ЭВМ."****// УСиМ. – № 6. – 1987. сс. 110-115.*

Аннотация. В статье рассмотрены назначение и принципы применения инструментальной расширяющейся системы программирования ТЕРЕМ. Опеределен класс языков, допускающих реализацию средствами системы. Программный базис системы используется в качестве ядра нового языкового процессора, синтаксическая подсистема которого строится системой ТЕРЕМ автоматически. Автоматизация построения семантических подсистем языковых процессоров основана на использовании пополняемого набора универсальных семантических процедур, реализующих наиболее употребительные в языковых процессорах структуры данных или выполняющих универсальные функции перевода.

Грудень 1987 *року Вченою радою Інституту кібернетики імені В.М. Глушкова мені було надано*

звання старшого наукового співробітника *за спеціальністю 05.13.11 – "Математичне і програмне забезпечення обчислювальних машин і систем".*
На цей час я мала стаж наукової роботи 14 років 10 місяців. Опублікувала 40 наукових праць, з них 23 – після затвердження кандидатської дисертації 12 травня 1974 року.
57

----------1988----------

16-18 лютого *відбувся II Всесоюзний н/т семінар* ***"Программное обеспечение многопроцессорных вычислительных систем". Учредители: Научно-производственное объединение (НПО) "Центрпрограммсистем",*** *Областное управление НТО Машпром, Калининский Дом техники НТО.*

Семінар проводився на базі НПО (конференц-зал турбази "Лисицкий бор").
Доповіді були розподілені по трьох секціях:

Секция 1. Вычислительные средства и ОС многопроцесорных комплексов.

Секция 2. Языки параллельного программирования. Разработка и внедрение прикладного программного обеспечения.

Секция 3. Аппаратные реализации программных средств. Методы и системы проектирования
аппаратных средств.

На першій секції були представлені доповіді з ІК АН УРСР:

Гороховский С.С. ***Распределенная операционная система МВК****.*

Фелижанко О.Д., Щеголева Н.Н. .

Дорошенко А.Е., Морозов С.И. ***О повышении отказоустойчивости макроконвейерных программ.***

*Федюрко В.В****. Доступ к внешней информационной среде в ОС ЕС 1766.***
Із 26 доповідей, представлених на Секції 1,13 доповідей було присвячено експедиційному геофізичному обчислювальному комплексу (ЕГОК) ПС-2000 та його удосконаленим модифікаціям. Сумарна швидкість ЕГОК з двох мультипроцесорів складала 400 млн коротких операцій за 1 сек. і 120 млн операцій додавання чисел з плаваючою комою. ЕГОК виготовлявся з 1981 року, 46% їх виробництва припадало на сейсморозвідку, решта – на розв'язок науково-господарських задач.

На Секції 2 були представлені доповіді:

Берестовая С.Н., Годлевский А.Б., Мищенко Н.М. Система параллельного программирования МАЯК

Молчанов И.Н., Рябцев В.Е. Об опыте практической реализации параллельных алгоритмов в задачах линейной алгебры.
58

Крат С.П. Многоуровневое динамическое распараллеливание последовательных программ.

Доповідь Молчанова І.М. відрізнялася від решти доповідей про високопродуктивні системи

конкретними прикладами роботи нашої МВК, які наводилися в доповіді.

На цій секції була проголошена доповідь

Миренков Н.Н. ***"Архитектура и математическое обеспечение вычислительной системы "Сибирь****".*

У ній розглядалися концептуальні основи проектування високопродуктивної обчислювальної системи, що створювалась в ОЦ Сибірського відділення АН СРСР для програмно-апаратної підтримки комплексного центру моделювання. Подальша доля цього проекту мені невідома.

18 березня. *Вдруге виступала офіційним опонентом на захисті кандидатської дисертації Подсвирова Володимира Миколайовича на тему "Исследование и разработка автоматизированных систем обработки данных, ориентированных на проектирование вычислительных систем".*

26-30 вересня відбувся VIII Всесоюзний семінар "Параллельное программирование и высокопроизводительные структуры" *(м. Алушта, турбаза "Прометей"). Учредители: АН СССР, Госкомитет СССР по науке и технике, Научно-технический институт межотраслевой информации при ГКНТ СССР, АН УССР, Ин-т кибернетики им. В.М. Глушкова, Симферопольский госуниверситет.*

На пленарних засіданнях виступили:

Ющенко Е.Л. "Структурированность параллельных вычислений: теория, инструментарий, сфера приложения";

Подловченко Р.И. "Проблема эквивалентных преобразований для схем программ с монотонными и перестановочными операторами";

Редько В.Н. "Прагматические аспекты композиционного программирования";

Летичевский А.А. "Параллелизм и решение задач на математических моделях предметных областей";

Нариньяни А.С. "Параллелизм и обработка знаний";

Вельбицкий И.В. "Технологические аспекты разработки параллельных программ";

Андон Ф.И. "Оптимизация вычислений в системах обработки данных".

Доповіді інших учасників семінару були розподілені між трьома секціями:

1. Методы параллельного программирования

2. Математическое обеспечение мультипроцессоров

3. Экспертные системы и прикладное параллельное программирование

В кожній секції було кілька підсекцій. Наша доповідь була запланована в підсекції

2.1. Макроконвейерные и распределенные вычислительные комплексы.

Мищенко Н.М., Федюрко В.В., Фелижанко О.Д. Интерактивная синтаксическая отладка программ в языках семейства МАЯК (с.110-111).

Аннотация. Языки семейства МАЯК – основные языки программирования для многопроцессорных вычислительных комплексов с макроконвейерной организацией вычислений (МВК). Комплекс средств общесистемного МО МВК, предназначенных для отладки программ в этих языках, включает интерактивный синтаксический отладчик, основными функциями которого являются:

– нисходящий левосторонний синтаксический анализ входных текстов, управляемый синтаксическими таблицами;

– обнаружение синтаксических ошибок и их нейтрализация (интерактивная и пакетная).

Отладчик реализован в виде синтаксического процессора на базе РСП ТЕРЕМ путем объектами двух видов:

– процедурами, которые обеспечивают работу с экраном дисплея;

– синтаксическими таблицами языка отлаживаемых программ.

59

Для нейтрализации ошибки сообщение об ошибке вместе с текстом всего предложения отображается на экран для исправления, после чего подается на вход Анализатору повторно. Нейтрализация синтаксических ошибок в пакетном режиме основана на пропуске фраз с ошибкой до специальных символов, извлекаемых отладчиком из синтаксических таблиц.

Объем отладчика – 2500 операторов ПЛ-1 ОС ЕС ЭВМ, он требует 170К памяти.

Жовтень*. III Всесоюзна н/т конференція "Методы синтеза типовых модульных систем*

обработки данных", Кишинев.

Подана доповідь Мищенко Н.М.

була відхилена. Коротко зміст відхиленої доповіді.

Модуль реализован на языке ПЛ/1 ОС ЕС и предназначен для автономного применения и для включения в трансляторы в качестве синтаксической подсистемы.

Функции модуля:

– построение синтаксических таблиц по описанию грамматик во входном метаязыке модуля;

– левосторонний нисходящий синтаксический анализ с ограниченными возвратами, управляемый построенными таблицами;

– поиск и нейтрализация синтаксических ошибок;

– построение дерева анализа;

– обход дерева с целью выполнения семантических действий, символьные имена которых нагружаются на элементы грамматик.

Модуль использовался автономно для отладки грамматик, а также включен в несколько языковых процессоров различного назначения, входящих в состав программного обеспечения МВК ЕС 1766.

***2 листопада**. Виступ на семінарі Наукової ради АН УРСР з проблеми "Кібернетика". Секція 1. Семінар 1.1. Теорія автоматів та її застосування.*

*Н.М. Мищенко. **"Об одном универсальном синтаксически управляемом трансляторе"***

*Н.Н. Щеголева **"Реализация в ОС ЕС программных модулей, взаимодействующих с макроконвейером"***

О.Д. Фелижанко "Интерактивное управление задачами в МВК"

Короткий зміст моєї доповіді. В докладе рассматривается универсальный транслятор, который включает синтаксический модуль, выполняющий синтаксически управляющий перевод класса языков, описываемых нелеворекурсивными контекстно-свободными грамматиками. Излагаются методика настройки транслятора на конкретный входной язык, алгоритм синтаксически-управляемого перевода и пути повышения его эффективности.

За текстом доповіді вийшла стаття у Збірнику ІК, затвердженому до друку 16.06.1989.

*Мищенко Н.М. **"Об одном универсальном синтаксически управляемом трансляторе"** / **Программные и аппаратные средства многопроцессорных вычислительных комплекс**ов. Сб. науч. тр. – Киев: ИК им. В.М. Глушкова АН УССР, 1989. – С. 4-11.*

----------1989----------

***Червень.** Підсумкова стаття в ж. Кибернетика*

*С.Н. Берестовая, А.Б. Годлевский, С.С. Гороховский, Ю.В. Капитонова, А.А. Летичевский, Н.М. Мищенко **"О реализации языков семейства МАЯК для многопроцессорных вычислительных Комплексов с макроконвейрной организацией вычислений"** // Кибернетика. – 1989. - № 3, с.29-34.*

Аннотация. Излагаются основные решения по проекту реализации языков параллельного программирования семейства МАЯК, обусловленные их отличиями от языков последовательного программирования и получившие практическое подтверждение в системе параллельного программирования МАЯК для МВК ЕС.

***5-10 червня**. Школа-семінар "Адаптируемые средства программирования. Методы оценки трансляторов", організована кількома відомствами Молдавської РСР та Киргизським с/г інститутом. Школа-семінар працювала в м. Фрунзе.*

60

*Подано доповідь на тему **"Инструментальный комплекс для реализации расширяющихся языков программирования**", але участі в роботі школи-семінара не брала.*

У другій половині 1989 року РСП ТЕРЕМ була прийнята в ФАП (Фонд Алгоритмів і Програм). Для цього були підготовлені тексти програм, документація, експериментальні дані, список заінтересованих організацій тощо.

***26 липня 1989 р.** складено Акт про прийняття програмного засобу у ФАП АН УРСР.*

***7 грудня 1989 р**. одержала Довідку про включення РСП ТЕРЕМ в Державний ФАП.*

Подаю коротку характеристику програмного продукту з супровідних документів.

РСП ТЕРЕМ предназначена для разработки языковых процессоров на ЕС ЭВМ для класса языков программирования и подмножеств естественных языков, используемых в человеко-

машинных системах и описываемых нелеворекурсивными контекстно-свободными грамматиками. Препроцессоры могут допускать смешанные входные тексты, в которых фрагменты в основном входном языке перемежаются фрагментами в базисном языке, описываемом наборами ключевых слов предложений базисного языка.
Синтаксические таблицы строятся самой РСП ТЕРЕМ по описанию грамматики входного языка в метаязыке, близком к языку бэкусово-науровских форм, а семантические модули разрабатываются пользователями по предлагаемой в документации методике.

Принцип применения РСП ТЕРЕМ состоит в том, что она вместе с синтаксическими таблицами берется в качестве синтаксической подсистемы будущего языкового процессора, к которой подсоединяются семантические модули.
61

РСП ТЕРЕМ обеспечивает информационное окружение для семантических модулей и их связь с синтаксической подсистемой. Для облегчения разработки семантических подсистем создан комплекс поставляемых отдельно семантических модулей для выполнения наиболее часто встречающихся в языковых процессорах функций перевода.
Режим работы – пакетный. Время счета контрольного примера на ЕС ЭВМ – 4, 84 сек.

Объем программной документации – 187 стр.
Технические характеристики РСП ТЕРЕМ:

– язык программирования – ПЛ/1 ОС ЕС ЭВМ;
– машина – ЕС ЭВМ 1040 и выше;
– операционная система – ОС ЕС версия 6.1;
– режим работы – пакетный;
– время счета контрольного примера – 3 сек. на ЕС 1066;
– объем загрузочного модуля – 119 кбайт;
– объем оперативной памяти – 162 кбайт
– объем программы – 3260 операторов языка ПЛ/1.

***Структура программы**. Программа состоит из главной процедуры terem и 9-ти внешних процедур: import, scaner, constr, intsyn, analiz, basicl, sem, link, export.*

Структура РСП ТЕРЕМ изображается в виде списка зависимостей процедур. В начале строки
этого списка стоит имя процедуры, после нее в той же строке через запятую указаны имена дополнительных входов в процедуру, если они имеются. После имени процедуры и ее входов в скобках указаны имена процедур и входов, к которым процедура непосредственно обращается. Затем указывается функция процедуры.
Начинается список строкой с именем главной процедуры terem.
terem (import, inprog, constr, analiz, link)
terem обрабатывает входные параметры, инициализирует общие переменные, управляет остальными процедурами;
import, inprog (link)
import читает входной текст из входного файлаЃС имеет еще один вход inprog
scaner вызывается из остальных процедур при необходимости прочитать очередную лексему из входного текста;
constr (inprog, scaner, intsyn)
constr анализирует предложения метаязыка программы и интерпретирует их;
intsyn (scaner)
ntsyn превращает правила к-с грамматики в синтаксические таблицы;
analiz (inprog, basicl, scaner, sem)
analiz осуществляет нисходящий левосторонний синтаксический анализ программ во входном языке и строит дерево анализа;
basicl (scaner, inprog, export, link)
basicl распознает предложения базисного языка по их ключевым словам, используется в препроцессорах;.
Процедуры scaner, constr, intsyn, analiz, basicl образуют синтаксическую подсистему РСП.
sem (link)
sem осуществляет обход дерева анализа с целью выполнения ассоциируемых с деревом семантических действий);

link (export, finish)
link связывает имена семантических действий с выполняющими их процедурами. Эта процедура – единственная изменяемая при включении новых семантических программ в разрабатываемый языковой процессор;
export (finish)
export записывает в выходной файл результат трансляции, если результат – текст. Процедуры sem, link, export образуют семантическую подсистему РСП ТЕРЕМ в ее исходном состоянии. При использовании РСП для построения транслятора исходная семантическая подсистема расширяется за счет специальных переводящих программ.

***4 червня 1990 р**. програму розміщено у Державному реєстрі (див. нижче)*

62

Вітання від Дмитра Миколайовича Тодороя з Кишинева – останнє спілкування з керівником Робочої підгрупи з розширних мов програмування

63

----------1990----------

*Головна подія року – **випробування макроконвейєра ЄС 1766** державною комісією. Макроконвейєр – це 41 арифметичний процесор. Була поставлена задача "Моделювання ядерного вибуху". Машина працювала безперервно 34 години зі швидкістю 632 млн операцій за сек. Випробування пройшли успішно.*

Проте комісія звернула увагу на деяке зменшення швидкості наприкінці випробувального терміну. Як потім виявилося, кілька процесорів вийшли з ладу. Передбачаючи таку можливість нашої техніки з боку макроконвейєра, розробники Операційної системи забезпечили розподіл ролей відказних процесорів між працюючими. Віртуозна програмістська робота, яку не зумів би зробити той, хто ніколи не мав справи з ненадійною технікою.

***У березні** цього року була роздрукована документація про "Расширитель" – препроцесор для трансляторів з контекстно-вільних мов, реалізований за допомогою РСП ТЕРЕМ. Об'єм власне "Расширителя" – 344 оператори ПЛ/1.*

***Анотація.** Расширитель предназначен для введения в языковой процессор текстов для замены и составных частей семантических определений языка-расширения, а также для организации многоуровнего перевода объектов языка-расширения, семантическое определение которых содержит тексты для замены, имеющие описанную в данном документе структуру. Документ является дополнением к документации РСП ТЕРЕМ.*

***16-18 січня**. Всесоюзна конференція "Актуальные проблемы системного программирования. Объектно-ориентированное программирование". Таллин, Институт кибернетики ЭССР.*
З нашого Інститут було прийнято тези трьох доповідей:

А.А. Летичевский, Ю.В. Капитонова. "Объектно-ориентированный подход и параллельное программрование".

Г.Е. Цейтлин, Е.Л. Ющенко. "Объектное параллельное программирование. Математические основы, инструментарий, сфера приложений".

Т.А. Валькевич "Объектно-ориентированный подход к построению словаря мультимодульных программ".

*Представлені мною тези доповіді "Об одном комплексе программных модулей для сборки трансляторов" були відхилені: результат непроханого вторгнення у 1983 році чи недоліки тез? Подаючи тези знову і знову, я тоді не пов'язувала відмову з моїм непроханим вторгненням. Аж тепер я прийшла до цього висновку, а якби "прозріла" вчасно, то не турбувала б оргкомітет. Зважаючи на те, що у тезах викладено **підсумок** результатів програмування за 1980-ті роки, вважаю за потрібне навести склад програмного комплексу, описаний в тезах.*

Программный комплекс содержит модули, в которых реализованы хорошо изученные и часто встречающиеся в трансляторах функции (функциональные модули) и структуры данных (объектно-ориентированные модули). В комплексе имеется главный модуль (модуль-ядро), содержащий Анализатор, который допускает класс языков, описываемых нелеворекурсивными контекстно-свободными грамматиками. Остальные модули подчинены главному. Каждый модуль представляет собой процедуру или пакет процедур и снабжен методикой и средствами его привязки к другим модулям.

Сколь угодно большой набор готовых модулей не исключает создания пользователем

непредусмотренных в комплексе модулей или "своих" версий имеющихся модулей по прилагаемой к комплексу методике.

Основные модули комплекса:

– модуль-ядро, кроме синтаксического анализа выполняет нейтрализацию синтаксических ошибок, построение дерева анализа, обход дерева с целью выполнения действий, имена которых нагружены на узлы дерева;

– модуль-связка имен действий с соответствующими процедурами;

– модуль, выполняющий интерактивную синтаксическую отладку;

– словарный модуль и др.

Комплекс был разработан в рамках математического обеспечения МВК с макроконвейерной организацией вычислений ЕС 1766 на языке ПЛ/1 ОС ЕС. Он послужил базой для разработки шести трансляторов различного назначения, функционирующих на ЕС ЭВМ.

64

1990 рік був роком закінчення кількох тем. Перелік частини тих тем, у яких я брала участь, наводиться на початку опису подій 1986 року, коли ми почали їх виконувати. Протягом 1990 року було написано кілька звітів. Наводжу назви тих, копії яких збереглися.

***16 травня** представила звіт за темою Лингвист – УН 2Г.100. 05. Б: "Инструментальные средства для построения диалоговых систем анализа и оценки ситуаций при планировании контроля текущей деятельности по разработке языковых процессоров" – (заключительный отчет).*

п.2. Инструментрий по созданию языковых процессоров (16 стр.).

***8 червня** подала Рисцову І.К. звіт за темою Лава – 311. III 1.100.07*

*Назва звіту: **"Исследование принципов построения и разработки семиотических систем***

***управления с использованием искусственного интеллекта"** (закл.).*

п. 2.3. Языковые процессоры РВС (Распределенные выч. системы) – стр. 51-61.

***19-26 червня** в Уфі відбувся X Всесоюзний семінар "Параллельное программирование и высокопроизводительные системы: методы представления знаний в информационных технологиях". Учредители: АН СССР, Госкомитет СССР по науке и технике, Научно-технический институт межотраслевой информации при ГКНТ СССР, Институт математики Уральского отделения АН СССР, АН УССР, Институт кибернетики имени В.М. Глушкова, Уральский авиационный институт им. С. Орджоникидзе, Уральский филиал ВИНИТИ статинформ.*

Доповіді розподілені по 4-х секціях:

1. Теоретические аспекты параллельного программирования

2. Технологические и инструментальные средства разработки программ

3. Представление знаний и экспертные системы

4. Вычислительные системы параллельного программирования.

Наша доповідь в Секції 4 (доповідач Федюрко В.В.).

*Т.А. Валькевич, Н.М. Мищенко, В.В. Федюрко, Н.Н. Щеголева **"О внутреннем представлении мультимодульных программ"***

Внутренее представление мультимодульных программ в МВК с макроконвейерной организацией вычислений представляет собой дерево статической подчиненности (ДСП) программных модулей ММ-программы, нагруженное информацией, необходимой для трансляции, запуска, выполнения и сопровождения этой программы. Внутреннее представление генерируется на основе входного текста ММ-программы и является основным объектом его информационной среды.

Статическая подчиненность программных модулей ММ-программы определяется вложенностью описания одних модулей в другие или явным указанием подчинения, если модули описываются раздельно. Самый внешний модуль, содержащий описания остальных, подчиненных модулей называется главным. Информация, собранная в ДСП, позволяет осуществлять независимую трансляцию простых программных модулей и их взаимодействие в процессе выполнения.

25-29 червня** в **Кишиневі відбувся Всесоюзний семінар "Адаптируемые средства программирования". Учредители: Кишиневский сельскохозяйственный институт им. М.В.

Фрунзе, Молдавское республиканское правление союза НИО СССР, *Дом науки и техники МРП СНИО СССР. Тези доповіді надіслала без участі в роботі семінару. Н.М. Мищенко "Об адаптации программных модулей, предназначенных для сборки языковых процессоров".*

Актуальность задачи автоматизации разработки языковых процессоров подтверждается не т олько появлением новых языков программирования, но и необходимостью адаптировать уже существующие языки к конкретному применению. Свойство расширяемости языков и соответствующих систем программирования является предпосылкой для адаптации реализованных систем программирования к использованию в новых условиях. В РСП ТЕРЕМ предлагаются следующие средства адаптации:
65

1. Автоматическое порождение синтаксических таблиц входного языка;

2. Адаптация как процесс развития входного языка, в роли которого предусматривается класс языков, описываемых модифицированными БНФ;

Через день після закінчення семінару у Кишиневі у Владивостоці розпочалось робоче засідання з системного математичного забезпечення, яке проводилося на базі Інституту автоматики і процесів управління Далекосхідного відділення АН СРСР у м. Владивосток 1–7 липня 1990 р. У програмі планувалися доповіді і дискусії на теми: цільові підгрупи, розширні системи, оцінки трансляторів, стандартизація мов програмування. З інформаційного листа: "Для участия в работе семинара Вам необходимо до 15 мая прислать оргвзнос 15 рублей (+ деньги на оплату дополнительных экскурсий, если Вы желаете принять в них участие).

Предполагаются следующие экскурсии: по Амурскому заливу – 3 руб., в морской заповедник на острове Попова – 7 руб., в г. Находку – 15 руб. Итого, максимальная сумма перевода – 40 руб."

Председатель Рабочей группы по ***языкам и системам программирования Комиссии СМО ОИВТА АН СССР И.В. Поттосин****.*

На жаль, я не брала участі у засіданні Робочої групи у Владивостоці.

У 1990 році у нашому лексиконі з'явилося нове слово трансп'ютер.

6 липня *від свого імені, від імені Щоголевої Н.М. та Валькевич Т.А. я подала інформацію про засоби програмного забезпечення трансп'ютерної системи, яку почали розробляти у відділі. Поскольку архитектура общесистемного МО обычно включает языковые процессоры (препроцессоры), считаем неизбежным разработку последних в рамках системы программирования многопроцессорных комплексов. Предлагается применение принципа сборочного программирования языковых процессоров, доказавшим свою эффективность в предыдущих разработках. Для этого могут быть использованы следующие результаты:*

1. ***Система сборочного программирования (ССП)****:*

– комплекс программных модулей (РСП ТЕРЕМ) на ПК (Мищенко, Щоголева);

– словарный модуль на ЕС 1766 (Валькевич).

2. Архитектра системы программирования. Принцип декомпозиции мультимодульных программ (Мищенко).

3. Методика использования ССП (Мищенко).

Термін трансп'ютер використовувався _____недовго, здається, ми не перенесли наші напрацювання для Макроконвейра на транспютери, хоча й починали цю роботу.

Премія*. За другий квартал 1990 року ми одержали премії за виконання таких робіт: Разработать проект ВИС и оболочку для управления процессом решения задач на транспютерных системах (Щеголева Н.Н., Фелижанко О.Д.).*

Разработать проект системы сборочного программирования для ТМ ЭВМ (Мищенко*)*

Премія*. За третій квартал нам дали премію за такі роботи : инструментальные средства для адаптации системы сборочного программирования ко входным языкам ТМ ЭВМ (Щеголева Н.Н.). Разработать методику применения системы сборочного программирования для реализации языковых процессоров ТМ ЭВМ (Мищенко Н.М.)*

13 вересня *представила звіт про виконання теми С.Г.100.01 "Создать и ввести в опытную эксплуатацию в ИК им. В.М. Глушкова АН УССР САПР*

высокопроизводительных ЭВМ с аппаратной реализацией системного программного обеспечения на основе перспективной элементной базы":
Назва звіту: "Принципы системной технологии в проектировании высокопроизводительных ЭВМ с аппаратной реализацией системного программного обеспечения на основе перспективной элементной базы".

п1. Сборочное программирование языковых процессоров (8 стр.).

66

п2. Инструментальная система сборочного программирования языковых процессоров (8стр.)

***19-20 вересня** у Дніпропетровську відбулась 2-а Всесоюзна науково-технічна конференція "Практическое применение современных технологий программирования, пакетов прикладных программ в вычислительных системах и сетях". Учредители: Госкомитет СССР по науке и технике, Минэлетротехприбор СССР, Днепропетровское научно-производственное объединение "Орбита", Днепропетровское областное правление НТО приборостроителей им. С.И. Вавилова.*

На конференції працювали 4 секції:

Секция 1. Применение современных технологий программирования

Секция 2. Практическое применение пакетов прикладных программ в вычислительных системах

Секция 3. Программно-математическое обеспечение локально-вычислительных сетей и микропроцессорных систем

Секция 4. Применение функциональных пакетов прикладных программ для народного хазяйства.

На секції 2 були проголошені дві наші доповіді:

Валькевич Т.А. "Пакет процедур для обработки словарей мультимодульных программ".

*Мищенко Н.М. **"Об одном комплексе программных модулей для сборки трансляторов.***

***Аннотация другої доповіді.** Рассматриваемый комплекс предназначен для реализации однопроходных трансляторов на ЕС ЭВМ для класса языков, который включает языки программирования и подмножества естественных языков, используемых для общения в человеко-машинных системах и описываемых нелеворекурсивными контекстно-свободными грамматиками.*

В программных модулях комплекса реализованы хорошо изученные и наиболее часто встречающиеся в трансляторах функции перевода и структуры данных. Предполагается участие пользователя в создании непредусмотренных в комплексе модулей или "своих" версий уже имеющихся модулей по прилагаемой к комплексу методике.

Дніпропетровськ. Історичний музей. Фото О. Кузьменка, 1968 р.

У Дніпропетровську сподобався Історичний музей, заснований знаменитим істориком Дмитром

Яворницьким. Вразила ширина Дніпра. Спочатку було здивування, наступної миті пригадала: Дніпропетровськ розташований над Дніпром нижче Києва.

67

***Жовтень.** Стаття в журналі УСиМ*

Н.М. Мищенко "Средства расширения входных языков РСП ТЕРЕМ и их применение"// УСиМ. – 1990. – № 5. – С. 55-62.

***Аннотация.** В статье рассмотрены языковые и программные средства одноуровневых и многоуровневых расширений контекстно-свободных языков, входящие в состав РСП ТЕРЕМ и наследуемые каждым языковым процессором, реализуемым на ее базе. Система допускает также класс исходных базисных языков, описываемых наборами ключевых слов.*

***Листопад.** Стаття у Збірнику ІК, підписаного до друку 20.11.1990.*

*Мищенко Н.М. **О сборочном программировании языковых процессоров / Сб. Интеллектуализация***

***программного обеспечения информационно-вычислительных систем.** Сб.науч. тр. – Киев : Ин-т кибернетики им. В.М. Глушкова АН УССР, 1990. – С. 45-52.*

***Аннотация.** В работе рассматривается способ реализации языковых процессоров путем их сборки из готовых программных модулей, которые вместе со средствами и методикой их*

адаптации к конкретному реализуемому языку образуют систему сборочного программирования языковых процессоров.

Підведемо підсумки опису Теремківського періоду 1970-80-х років.
Головним моїм заняттям протягом цього періоду була побудова інструментальної системи програмування під назвою розширна система програмування (РСП) ТЕРЕМ та її застосування для розробки трансляторів для Макроконвейєра з мов, граматика яких описувалася мовою БНФ.

РСП ТЕРЕМ – це універсальна синтаксична підсистема (Аналізатор вхідних мов і Конструктор
синтаксичних таблиць) та універсальні програми семантичної підсистеми, відкритої для розширення. У транслятор, який розроблявся для мови з допустимого класу мов, повністю включався Аналізатор, а по БНФ-опису граматики мови програма Конструктор будувала синтаксичні таблиці, які з Аналізатором складали синтаксичну підсистему транслятора.
У складі семантичної підсистеми виділялися засоби систем побудови трансляторів (СПТ). Їх наявність у складі новоствореної системи програмування дозволяла програмістові змінювати семантику вхідної мови в процесі використання транслятора. Відсутність засобів СПТ в трансляторі означала, що вхідна мова фіксована і не потребує розвитку під час її використання. Можливий третій варіант: включення засобів СПТ у транслятор для розвитку вхідної мови до певного рівня, після чого засоби СПТ можут бути вилучені з транслятора.
Оригінальним був зв'язок синтаксичної та семантичної підсистем: до елементів опису синтаксису мовою БНФ дописувалися імена відповідних семантичних дій з ознаками, які регулювали час ініціації дій над фразою, розпізнаною під час синтаксичного аналізу.
Здатність вхідної мови до змін була необхідна для реалізації нових мов, що еволюціонують в процесі їх використання, як це було у випадку з мовами програмування для Макроконвейєра.
Частина виступів на семінарах та конференціях стосувалася загального опису використання

РСП ТЕРЕМ, інша частина висвітлювала окремі властивості трансляторів, побудованих за Д омогою РСП ТЕРЕМ, завдяки яким вдавалося реалізувати:

- *дослідницький характер реалізації мови, якщо в її транслятор включалися засоби СПТ;*
- *адаптовність системи до конкретних умов обчислювальної системи;*
- *практичність системи, оскільки 3/5 нового транслятора – це засоби РСП ТЕРЕМ.*

Протягом 1990 року пощастило перекласти РСП ТЕРЕМ, запрограмовану мовою ПЛ/1, на мову Сі для роботи на персональних комп'ютерах (ПК). Переклад мовою Сі виконувався буквально оператор за оператором, опис за описом. Яке ж було моє здивування, коли на першому ж сеансі роботи на ПК РСП ТЕРЕМ запрацювала!

Насамкінець виражаю щиру подяку колезі Феліжанко Ользі Дмитрівні за редагування та виправлення помилок на заключному етапі підготовки цього тексту.

Р а з д е л 2 Технология программирования по Глушкову до и после развала СССР (1960-1913)

К 90-летию и 100летию академика Глушкова В.М.

Программирование для первых компьютеров в СССР

С момента появления первых ЭВМ и языков программирования стали разрабатываться программы для решения разных задач с математики, алгебры, комбинаторики и т.п. на электронных машинах, авторство первой из них принадлежат С.А.Лебедеву (1957) . Период

(1958-1975) – включал разработку и создание разных ЭВМ (МЭСМ, М-20, Проминь, Урал, Днепр, Днепр-2, М-50, БЭСМ, Стрела и др.). Основные из них разрабатывались в ИК АН УССР под руководством В.М. Глушкова. Он инициировал много новых государственных программ с разработки и конструирования вычислительных машин, а также общесистемного программного обеспечения (ОПО) для функционирования ЭВМ (ОС, трансляторы, редакторы и др.) и для решения различных научно-технических задач.

Серия машин (1975-1982) «Мир» реализована на идеях встроенного ПО, реализующего не только общие системные функции ЭВМ, а и элементы доказательства программ, выполнения алгебраических преобразований и вычислений отдельных задач математическими методами. И наконец, отечественная рекурсивная машина «Маяк». успешно объединила в себе наработанные идеи построения ЭВМ и новую идею интеграции машин и систем, выполнена на высоком научно-техническом уровне, не имеющая аналога.

В плане создания на машинах систем автоматизирующих организационно-производственную деятельность разных сфер управления предприятиями с помощью ЭВМ В.М.Глушкову принадлежит идея создания АСУ и ОГАС. АСУ предприятиями было создано много в СССР, а вот ОГАС не удалось привести в жизнь из-за непонимания руководящих органов страны и нежелания предоставить финансовый ресурс на реализацию проблематики всесоюзной распределенной сетевой обработки.

Центральной стратегической задачей производства компьютеров и их общесистемной поддержки является технология программирования (ТП).

Первая конференция по программированию была проведена в Киеве (**1975)** в зале Октябрьского Дворца, в которой принимали участие видные специалисты: ЕЛ. Ющенко (ИК АН УССР), М.Р.Шура-Бура, Э.З Любимский (ИПМ им. Стеклова), А.П.Ершов (ВЦ Новосибирск), Гангадзе Е. (Грузия), Фуксман (Ростов, РГУ), Терехов А.В. (Ленинград, ЛГУ) и др.

Главный вопрос этой конференции – теория и практика составления трансляторов в ЯП, которые предлагались специалистами разных академических и республиканских институтов СССР.

Повышенный интерес к программированию был у Глушкова в связи с разработкой системных программ функционирования новых ЭВМ, привнесших в сферу программирования новые методы алгебраического преобразования математических вычислений на машинах «Мир», а также технических средств синтаксической проверки правильности программ (РТК-комплекс) Велькицкого И.В и АПРОП (Глушков В.М., Лаврищева Е.М.). С этих работ (1967) начался период нового направления в программировании - технология. Кандидатскую диссертацию по ней Велькицкий И.В. защитил под руководством Глушкова и Ющенко Е.Л. Потом параллельно с работами, которые возглавляла в институте кибернетики Е.Л. Велькицкий И.В., занял лидирующую роль в этой области и потеснил ее.

По инициативе Глушкове в ГКВТИ была создана научно-техническая комиссия по ТП (1980), в которую возглавил от ИК Велькицкий И.В. С его участием была разработана комплексная программа научно-технического прогресса стран – членов СЭВ до 2000года программы 1.1.6. “Развитие технологии разработки и промышленного производства программных средств вычислительной техники (1981г.). Глушков поддержал создание Международного Центра Технософт при ИК АН УССР (1980). Коллектив центра выполнил большое количество работ в области технологии программирования и автоматизации производства программ с использованием технических средств анализа и контроля программ. Центр Технософт провел две конференции - всесоюзную и международную по технологии программирования (1986, 1992).

Было сформировано термин – технология программирования – ТП.

Одной из главных направлений конкурса Интерфейс СЭВ. Сотрудники отдела ЛЕМ участвовали с определением понятия интерфейса. Выставленная работа получила почетную грамоту по интерфейсу модульному, межязыковому и технологическому.

На 2-всесоюзной конференции 1979 «Технология программирования» академик А.П.Ершов в докладе по перспективам до 2000 г.выделил три поколения интегральной промышленной технологии программирования. Приведем отдельные аспекты этих направлений.

«Первое поколение (организационное программирование) 1975-1985 гг.

Язык программирования не формализован. Переход от прототипа к программной версии не формализован. …

Языки программирования - ФОРТРАН, КОБОЛ, ПЛ/1, Ассемблер.

База знаний отсутствует…

Развитие продукта – версионное…

«Второе направление (Сборочное программирование) 1985 – 1995гг.

Язык спецификации регламентирован.

Переход от от протопипа к промышленной версии регламентирован.

Язык разработки - формализованный язык высокого уровня со средствами модуляризации.

Язык программирования объединен с языком разработки, допуская комплексирование с ассембрерными модулями….».

«Третье направление (доказательное программирование) 1995 – 2005 гг. …

Язык разработки формализован и содержит систему формальных преобразований…

Язык программирования объединен с языком разработки,…

База данных проекта машинизирована.

Применение ЭВМ в проекте – полное.

Развитие продукта – эволюционное – адаптивное …».

Как и многие в ИК, я уважала В.М. Всегда слушала его выступления на семинарах, конференциях, ученом совете и т.п. В его выступлениях всегда было много новых идей, (ближних и дальних целей), которые подхватывали заинтересованные специалисты и начинали новое направление либо развивать с новыми идеями.

Многие и я ходили на конференции и семинары в ИК со школьной тетрадью (100л.) и записывала интересные для меня мысли, особенно, если они касались вопросов программирования или направлений, связанных с развитием новых машин, например, «Мир», Днепр и др.

Одним из судьбоносных памятных выступлений В.М. – доклад на семинаре 5 марта 1975 года в отделе под номером 100, посвященном представлению перспектив программирования в нашей стране, которые он сделал после конгресса IFIP.

На этом семинаре присутствовали видные специалисты ИК - Ющенко Е.Л., Сергиенко И.В., Летичевский А.А., Капитонова Ю.В., Молчанов И.Н., Лаврищева Е.М., Никитин А.И., Бабенко Л.П. и др. Глушков сказал: «Пройдет 20-30 лет сложные программы будут выпускаться, как на конвейере Форда из готовых «деталей». Появятся фабрики программ, работающие про принципу деталей в автомобильной промышленности Форда». Он верил, что выпуск программной продукции в перспективе будет проводится на на индустриальной основе.

На той час уже складывались такие п**редпосылки индустрии программ:**

1. Программная инженерия (ПИ) – качество, продуктивность, индустрия (1968).
2. Постановление ГКНТ СМ СССР – о создании Госта и РФАП (1964).
3. Постановление КМ СССР «Программные средства как продукция производственно-технического назначения» (1974г.).
4. А.П.Ершов – Готовые программы – чистая экономия труда в разработке больших программных систем (1978).
5. Программостроительный завод И.Карасева (1982).
6. Государственный Комитет по науке и технике – ГКНТ (1980г.).
7. Государственный Комитет СССР по вычислительной технике и информатике (1980): Совет по применению средств ВТ МПК ПО ВТ; Научно-техническая комиссия по технологии программирования.

8. Комплексная программа Научно-технического прогресса стран – членов СЭВ до 2000года (1983).
9. Научно-производстенное объединение «ЦентрПрограммСистем» (1980).
10. I - всесоюзная конференция «Технология программирования» (1979).
11. Всесоюзная научно-техническая конференция «Программные средства как продукция прозводстенно-технического назначения» (направления: Технология разработки программных систем» (1985).
12. Международная научно-техничная конференция - Интерсофт (Индустрия ПО, Новая информационная технология, Развитие теории и методов создания СУБД и др.), Центпрограмсистем – 1987г.
13. Международный конкурс по проблеме 1.1.6 “Развитие технологи разработки и промышленного производства программных средств вычислительной техники (ГК ВТИ, 1985г.).
14. II-всесоюзная конференция “Технология программирования” (1986), ГКНТ, ГКВТИ, АН СССР, АН Украинской ССР, Комиссия ГКНТ по технологи программирования).
15. 2-Всесоюзные конференции “Технология программирования 90-х” (МНЦТП Технософт, 1992).
16. Системы автоматизации производства программ (ПРИЗ, РТК, ПРОЕКТ, АПРОП, ДИСУППП, ДЕЛЬТАСТАТ, АЛЬФА,, БЕТА и др.. 1980-1992).

Основные направления ТП , способствующие становлению индустрии программ

1. Объектно-ориентированное, компонентное программирование, объектно-компонентное программирование (ОКМ) 2003г.
2. Сервисы (1996, CORBA).
3. SWEBOK, PMBOK (2001, 2004).
4. Типы данных ЯП – Ч.Хоар, Вирт (1974), В.Н.Агафонов (1981) …
5. Порождающее (Generative) программирование – семейство систем программ (2003).
6. Системы инструментальной поддержки ПС 1992-2012 (COM, CORBA, MS.VSTS, MSF), Eclipse, Protégé, Microsoft DSL Tools и др.).
7. ТЛ, ЖЦ, Product lines, стандарты ЖЦ, качества и др.
8. Система дисциплин ПИ поддержки индустрии программных продуктов
9. Типы данных общего назначения – Стандарт ISO/IEC 11404-1996, 2007.

Сразу же после выступления Глушкова В.М. на Ученом Совете многие отделы ИК занялись разработкой систем автоматизации, которые могли бы стать основой фабрик программ, работающих по принципу сборочного конвейера Глушкова. Эта идея плодотворно развивалась многие годы по разным направлениям в Институте Кибернетики: формализованные технические задания (Капитонова Ю.В., Летичевский А.А.); система автоматизации программ - АПРОП (Лаврищева Е.М.); комплексирование пакетов прикладных программ (ППП) для методов численного анализа (Молчанов И.Н.); блочное создания ППП статистики и оптимизации (Парасюк И.Н.); языковые спецификации для генерации систем обработки данных Макробол (Бабенко Л.П) и др.

По всей стране проводились конференции, в которых обсуждались разного рода решения в этом направлении. Возникли постановления правительства, ГКНТ и ГКНТИ проводились разные мероприятия, способствующие сделать программную продукцию, продукцией производственно-технического назначения.

Система АПРОП начала разрабатываться в НИЦЕВТ на ЕС ЭВМ

Система автоматизации производства программ включала такие понятийные термины – **объект, модуль, интерфейс, сборка, тестирование, отладка** и др.

Термин сборка программ стал популярным, его употребляли многие специалисты. Принцип конвейерной сборки программ из заготовок - КПИ, готовых программных элементов, предсказанный В.М.Глушковым широко применялся во многих организациях (Орлов В.И.).

Нам удалось построить теорию сборки, основу которой составлял интерфейс и методы преобразования типов данных, что передавались от одной программы к другой. В докторской диссертации ЛЕМ «Методы, средства и инструменты сборочного программирования СОД» (1989) отражены все аспекты сборки программных элементов. Была написана докорская диссертация – 351с. Талантливый программист Грищенко В.Н. разработал библиотеку интерфейса, которая содержала 64 функции преобразования типов данных применительно к языкам программирования (ЯП) четвертого поколения, реализованных в ОС ЕС. На эту тему Грищенко В.Н. защитил канд. диссертацию в 1989г.

Идея сборки модулей через модули посредники (связки) каждой пары разноязыковых модулей, механизмы генерации конкретных модулей посредников и библиотеки межъязыкового интерфейса (БМИ) опубликованы в трудах международных и всесоюзных конференций, и в монографии

Е.М.Лаврищева, В.Н.Грищенк В.Н. «Связь разноязыковых модулей в ОС ЕС» (М.: Финансы и статистика, 1982.-127с.. В этой книге в приложении 1-2 содержались механизмы использования библиотеки интерфейса при написании модулей связок двух разных модулей. В программисткой среде библиотека БМИ имела успех, ее мы передали в десятки организаций страны (по актам передачи), которые использовали ЕС ЭВМ для своих расчетов разных научно-технических задач. Она способствовала сокращению объема работ при сборки сложных комплексов программ из готовых разноязыковых программных элементов – модулей, компонентов, объектов..

С главным идеологом сущности библиотеки и механизмов сборки программ БМИ в системе АПРОП - Грищенко В.Н. после защиты диссертации Он защитил канд. диссертацию на эту была написана книга 1991году «Сборочное программирование» (Наукова думка, 213с.). И как оказалось, она используется студентами российских ВНЗ (в частности, МИФИ, МФТИ) в 2008году, когда я ехала в поезде Киев – Севастополь летом, для отдыха и в поезде был студент 5 курса МФТИ. Увидев книгу «Сборочное программирование» он рассказал, что в этом году ее использовал при работе над дипломом и сказал, как жаль, что машины ЕС ЭВМ перестали использовать ее преемница IBM-360, 370 до сих пор работает в США с большим числом ЯП (Smalltalt, Perl, …). Вопрос объединения модулей, реализован нами в системе АПРОП и в книге Связь разноязыковых модулей ОС ЕС ЭВМ. Было очень приятно услышать о жизни нашей книги Сборочное программирование от студента. Он сказал, жаль, что нет электронного варианта. Я его проинформировала, что готовиться переработанное издание. В этом издании включены три новых раздела: 5. Классификация и типизация программных ресурсов, 6. Объектно-компонентный метод разработки программ, 10. Основные положения и дисциплины сборочного производства программных продуктов.

Идея сборки оказалась долгоживущей. Это метод используется в разных современных операционных средах (VS.Net, Corba, IBM, Java и др.). Принципы ее реализации близкий в сборке. Опишем.

Главным объектом обработки новых ПС были готовые программы библиотек, представляющие собою стандартные машинные подпрограммы общего назначения, реализованные на конкретной ЭВМ в конкретных ЯП. Этот путь развития и совершенствования готовых компонентов повторного использования (КПИ) определялся возможностями вычислительной техники, операционных сред и достижений в области теории и практике программирования программных комплексов и систем. Одна из ключевых проблем современного программирования - КПИ, которыми могут быть модули, программы, алгоритмы, спецификации, и т.п., пригодные для использования при разработке новых ПС. В КПИ

материализуется многолетний опыт компьютеризации разных сфер человеческой деятельности, который может применяться непосредственно либо путем настройки и адаптации к новым условиям среды обработки. Первыми представителями КПИ, готовых к использованию был подпрограммы библиотек машинных программ (1960г.). Программирование с использованием КПИ обогатило метод проектирования (снизу-вверх) сложных программ из более простых. Оно прошло длинный путь своего развития от полуавтоматизированных библиотек до электронных библиотек и репозиториев.

В это время широко использовались большие ЭВМ (ЕС ЭВМ, БЭСМ-6, М-20 и др.), набженные компиляторами с ЯП: Ассемблер, Алгол, ПЛ-1, Фортран, Snobol, SmallTalk, Basic, Кобол и др.. Разнообразие ЯП и большой объем памяти ЕС явились базисом реализации идеи практической сборки программ на машинах ЕС ЭВМ.

На начальном этапе исследование проблемы сборки программ проводилось по трем направлениям.

1. Анализ средств создания крупных программных комплексов с монолитной и динамической структурой, а также особенностей объединения программ, отличающихся представлением типов и структур данных ЯП для их формализованного описания и определения стандартизованного задания объектов сборки, как это принято в промышленности. Исследовался и оформился базис теории абстрактных типов данных и подходов к формальному их преобразованию. На основе этой теории и теории алгоритмических алгебр Глушкова были исследованы все типы и структуры данных подмножества ЯП и построены алгебраические системы формального преобразования типов и структур данных ЯП, позволяющие установить взаимно однозначное соответствие данных каждой пары ЯП указанного подмножества. В результате были разработаны 64 функций преобразования одного типа данных к другому в подмножестве ЯП и определен специальный класс интерфейсных модулей, как механизмов связывания разноязыковых объектов.

2. Сформированное в практике программирования понятие - повторная подпрограмма библиотеки машинных программ (Курочкин, Москва) было изменено до уровня исходного представления на любом из ЯП в специальном хранилище - Банке модулей с функциональными разделами (вычислительная математика, экономика, АСУ и др.). Решение проблемы сборки разных КПИ в исходном представлении на ЯП из Банка модулей основывалось на разработке стандарта описания исходных модулей, который кроме описания тела модуля, включал описание его паспорта - информационного раздела, содержащего задание типов данных входных и выходных параметров и операторов вызова других модулей из Банка модулей. Определены новые принципы и механизмы стыковки исходных разноязыковых модулей, основанные на интерфейсных модулях. В этот период проблема КПИ решалась и на государственном уровне путем создания:

готовых алгоритмов и программ с бумажной документацией и на носителях (МЛ/ПК) и их складирование в республиканских и государственном фонде для последующего распространения и продажи их по стране;

программостроительного завода (фабрики) для сборки новых АСУ из готовых программных КПИ и заготовок;

Государственных программ проведения НИОКР при ГКВТИ СССР по разработке инструментальных систем автоматизации сложных программных комплексов и пакетов прикладных программ (АПРОП, ПРИЗ, ЯУЗА, СИГМАСТАТ, МАКРОБОЛ и др.).

Программы Фондов получили статус изделий или продуктов производственно-технического назначения согласно соответствующего постановления КБ СССР практически использовались по прямому назначению, в то время как вновь создаваемой системе их использование было затруднено из-за бумажного их представления, отсутствия формализованных интерфейсов с другими программами, необходимости проведения различного рода переделок.

Фонды просуществовали более 10 лет и автоматически перестали существовать, как только ушли со сцены ЕС ЭВМ.

Завод в Калинине проработал несколько лет. На нем было построено два опытных образца АСУ путем модернизации ранее сделанной. К тому времени в Фондах заготовок программ для АСУ было мало. Завод, как говорят сегодня, обанкротился из-за отсутствия: готовых «деталей» на его складе типовых изделий для АСУ; технологии стыковки используемых программ; необходимых ресурсов (оборудования, памяти, специалистов и др.); финансов и др.

3. Разработан метод сборки прикладных программ и систем, реализация которого позволило сформировать не только новый вид программирования – сборочное, но понятия – интерфейс, этапы жизненного цикла процесса сборки, моделирование функций автоматизируемой предметной области и разработка структуры системы сборки разноязыковых модулей и программ, получившей название АПРОП. Первая публикация по методу сборки модулей [1] зафиксировала основные пути реализации фабрики программ на машинах ЕС. Изготовленная система АПРОП-2 была передана в республиканский Фонд в Киеве, 1982г. и в ЕрНУЦ СНПО «Алгоритм», 1985г., а также внедрена в 23 организациях СССР. По системе опубликованы две работы (Связь разноязыковых модулей в ОС ЕС ЭВМ, М.:, «Финансы и статистика» 1982г., 127с. и «Сборочное программирование, Киев, Наукова думка, 1991, 286с.), защищены 2 кандидатские и докторская диссертации, получена премия СМ СССР (Лаврищева Е.М., 1985г.) и почетная грамота международного конкурса «Интерфейс СЭВ» 1987г.

Главным нововведением в концепции сборочного программирования является интерфейс (межмодульный и межязыковый). Его задачи сформулированы в 1976г. и намного опередили зарубежные разработки. В настоящее время интерфейс сохранил свою актуальность и выступает в качестве главной доминанты взаимодействующих компонентов и объектов в современных глобальных и сетевых средах.

Межмодульный интерфейс - это интерфейсный модуль-посредник между двумя взаимодействующими программными объектами, выполняя функции передачи и приема данных между ними. Впервые разработан язык определения интерфейсов (ЯОИ) для описания операторов вызова модулей, параметров и их типов, а также операций проверки правильности обмена данными.

Межъязыковой интерфейс - совокупность средств и методов преобразования структур и типов данных между ЯП с помощью алгебраических систем и функций (и макроопределений) библиотеки интерфейса для взаимно-однозначного обмена данными между разноязыковыми модулями через механизмы интерфейса. Библиотека интерфейса в момент широкого использования ЕС была передана в 50 организаций СССР для самостоятельного применения при работе с разными языками ОС ЕС [ ].

Созданная нами концепция интерфейса модулей, как средства связи разных типов объектов в ЯП путем автоматически генерируемого интерфейсного модуля-посредника, является первой отечественной парадигмой интерфейса в программировании, реализованной в 1975-1982г. в системе АПРОП. Гораздо позже в 1985-90 годах появились средства описания интерфейсов объектов за рубежом - языки API (Application Progpam Interface) и IDL (Interface Definition Language).

В основе реализованного нами сборочного программирования в системе АПРОП лежит метод сборки (интеграции) сложных программ и специализированных технологий программирования для классов задач обработки данных.

Метод сборки программ включает математические формализмы определения связей (по данным и по управлению) между разнородными объектами и генерация интерфейсных модулей-посредников для каждой такой пары объектов. Каждое взаимодействие описывается с помощью операторов вызова типа CALL с множеством фактических параметров, передаваемых вызывающему объекту.

Сущность решения задачи интерфейса пары разноязыковых объектов состоит в построении взаимно однозначного соответствия между множеством фактических параметров $V= \{v^1,v^2 ,...v^к\}$ вызывающего объекта и множеством формальных параметров $F = \{f^1, f^2, ..., f^{к1}\}$ вызываемого объекта и их отображения с помощью алгебраических систем.

Каждому типу данных T_α^t языка l_α ставится в соответствие алгебраическая система $G_\alpha^t = <X_\alpha^t , \Theta_\alpha^t >$, где X_α^t - множество значений рассматриваемого типа, а Θ_α^t - множество операций над объектами данного типа. Т.е. операции преобразования типов данных соответствует изоморфное отображение алгебраической системы G_α^t в G_β^d. Построен класс алгебраических систем $\Sigma = \{G_\alpha^b, G_\alpha^c, G_\alpha^I, G_\alpha^r, G_\alpha^a, G_\alpha^z\}$ и рассмотрены все виды преобразований типов данных (b-boolean, c-character, i-integer, r -real, a -array, z–record и др.). Доказан изоморфизм алгебраических систем и отсутствие для множеств X_α^t и X_β^q. Установлено, что мощности алгебраических систем равны $G_\alpha^t | = G_\beta^q$.

Структура АПРОП

Главные объекты системы АПРОП: модуль, модуль-посредник, межмодульный интерфейс, Банк модулей и метод проектирования систем снизу – вверх с выбором готовых модулей из Банка модулей и их сборки в новые программные структуры. Базовая концепция системы – интерфейс как аппарат связи ЯП и модулей, записанных в разных ЯП (Фортран, ПЛ/1, Алгол, Ассемблер, Кобол). Каждый исходный интегрируемый («погружаемый») в АПРОП прикладной модуль имел паспорт, в котором описывались сведения о назначении, объеме, параметрах и др. Среда была одна для всех – ОС ЕС.

Множество троек из вызываемого и вызывающего модулей в разных ЯП и модуля-посредника объединялись в системе АПРОП в агрегат – монолитный продукт на ЕС ЭВМ, предназначенный для решения класса прикладных задач. В функции посредника входило отображение формальных и фактических параметров, проверка соответствия передаваемых параметров (количества и порядка расположения), а также их типов данных. Типовая схема связи разноязыковых объектов показана на рис. 1.

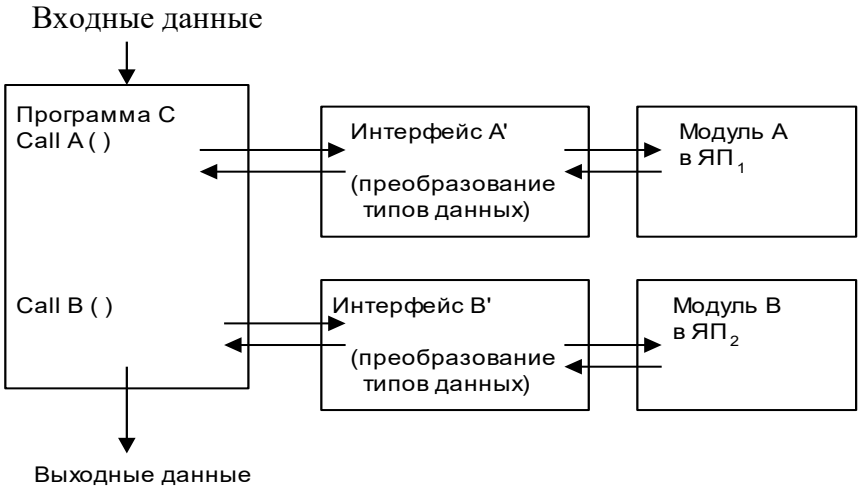


РИС. 1. Схема вызовов модулей А и через интерфейсы А’ и В’

На схеме приведена программа С, в которой содержатся два вызова – Call A () и Call B () с параметрами. Эти вызовы «проходят через» интерфейсные модули-посредники А’ и

B’, которые осуществляют функции преобразования данных и их передачу модулям A и B. После выполнения модулей A и B результаты преобразуются обратно к виду программы C. Если типы данных параметров – не релевантные (например, передается целое, а результат – вещественное или наоборот), то в функции посредника входит прямое и обратное их преобразование. Сгенерированный модуль-посредник содержит обращения к элементам библиотеки интерфейса, которые выполняются в момент перехода одного модуля к другому и обратно.

Общая структура системы АПРОП (рис. 2) включает элементы реализации сборки разноязыковых программ и следующие виды работ по сборке модулей:

– обработка паспортных данных модулей в ЯП;

– анализ описания параметров модулей и составление задания на обработку несоответствующих типов данных, проверка правильности передачи параметров как по их количеству, так и по соответствию типов данных в классе ЯП;

– преобразование типов данных в ЯП (b–boolean, c–character, i–integer, r–real, a–array, z–record и др.) в виде обращения к функциям библиотеки интерфейса;

– генерация модулей-посредников и составление таблицы связи пар компонентов;

– интеграция пар модулей и их размещение в структуре программного агрегата;

– трансляция и компиляция модулей в ЯП агрегата в виде готовой программной структуры;

– трассировка интерфейсов и отладка функций модулей в каждой паре агрегата;

– тестирование программного агрегата в целом;

– формирование программ запуска программного агрегата и документации.

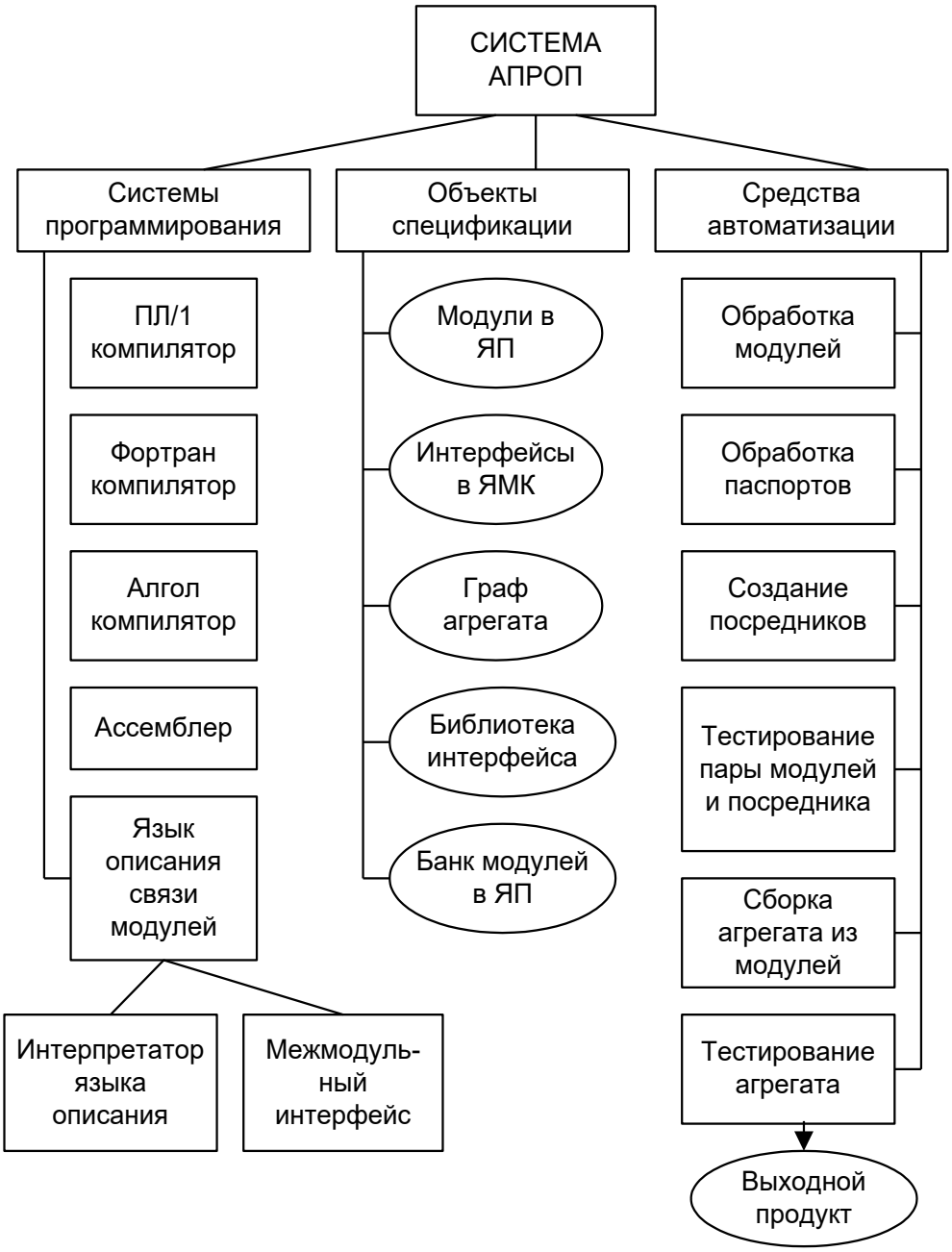


РИС. 2. Структура системы автоматизации производства программ

В системе реализовано два типа интерфейса – интерфейс модулей и пары ЯП, т.е. межмодульный и межъязыковой интерфейсы.

Система АПРОП реализована в среде ОС ЕС ЭВМ, принципы функционирования которой разработаны Е.И. Моренцовым, В.М. Зиньковичем. Она выполнялась по договору с Министерством радиопромышленности СССР как составная часть программных проектов «РУЗА» и «ПРОМЕТЕЙ» (В.В. Липаев). Система АПРОП апробирована для автоматизации взаимосвязи разноязыковых модулей с помощью модулей-посредников для языков четвертого поколения. Она передана в ЕрНУВЦ с документацией объемом более 1000с., внедрена в 52 организациях СССР (в Москве, Ленинграде, Минске, Риге, Ереване и др.) и отмечена Премией

Совета министров СССР (1985 г.) по указанному министерству с включением двух авторов (Е.М. Лаврищева, А.И. Никитин).

Теоретическое обобщение концепции и метода в системе АПРОП – *теория сборочного программирования*, защищена в докторской диссертации Лаврищевой Е.М. («Модели, методы и средства сборочного программирования», 1989), кандидатской диссертации Грищенко В.Н. («Реализация межмодульного интерфейса в системе АПРОП», 1991) и отображена в монографиях [5, 6].

Таким образом, *интерфейс модулей* как средство связи разных типов объектов в ЯП – первая отечественная парадигма интерфейса в программировании практически реализована в системе АПРОП (1974–1985 гг.). Интерфейс развивался за рубежом в проектах MIL, SAA, OBERON для комплексирования модулей на разных вычислительных машинах. В настоящее время идея связи программ через интерфейс для класса современных языков описана в [16].

Метод сборки программных объектов использовался при реализации:

системы автоматизации производства программ (АПРОП);

программно-технологического комплекса ПРОГРАММА-ПРОГРАММА , выполняемого (1987-1989 гг.) по линии КП НТП СЭВ «ИНТЕРФЕЙС» и обеспечивающего интеграцию программ (отв. Моренцов Е.И.);

системы АПФОРС, автоматизирующая создание пакетов прикладных программ (ППП), ответственный Хоролец Д.С. (1989-1991);

экспертной системы, информирующей возможности Банка модулей и обеспечивающей подбор необходимых объектов, как КПИ для решения задачи или использования в новой системе (диссертационная работа Юрия), который сразу после образования независимой республики Украины, бросил науку и пошел работать в Банк.

Система АПРОП стала прототипом для построения новой системы обеспечения взаимодействия разноязыковых программ в классе современных ЯП, новых компьютеров и сред. Новая особенность такой системы – наличие трех языков описания интерфейсов API, IDL и RMI для представления разных видов модулей-посредников в ЯП и соответствующих сред их выполнения.

Прикладным базисом взаимодействия программ может стать руководство [16], где автор Гринфилд представил более 100 вручную составленных вариантов модулей-посредников в классе современных языков: C/C++, Visual C++, Visual Basic, Matlab, Smalltalk, Lava, LabView, Perl. Эти варианты практически проверены автором в современных средах функционирования. Среди них не учтены проблемы, связанные с появлением новых современных архитектур компьютеров и сред. Варианты связей можно было бы доработать и реализовать новый проект современной системы автоматизации взаимодействия разноязыковых программ.

Таким образом, концепция интерфейса модулей, представленная в парадигме сборочного программирования, внесла значительный вклад в развитие теории программирования сложных систем из программ, разработанных средствами современных ЯП, инженерных подходов к конструированию и управлению КПИ. Роль этой концепции возрастает благодаря накопленному огромному запасу КПИ, т.е. reuse-компонентов в разных предметных областях, использовать которые без определения интерфейса не представляется возможным.

2.2. Развитие идеи связи языков и программ в распределенных средах

С годами проблема связи разноязыковых, разнородных (по коду и среде) программ обострилась в связи с быстрым изменением архитектуры компьютеров, появлением распределенных, клиент-серверных сред и т.п. Проявилась неоднородность ЯП как в смысле представления в них типов данных, так и платформ компьютеров, на которых реализованы соответствующие системы программирования, а также в различных способах передачи

параметров между объектами в разных средах – маршаллинг данных через разные виды операторов удаленного вызова. Единого подхода к решению проблемы интерфейса не существовало. Стандарт ISO / IEC 11404-96 определил подход к решению вопросов интерфейса всех видов ЯП с помощью универсального языка LI (Language Independent), независимого от ЯП. Однако инструментальной его поддержки до настоящего времени не существует. Пользователям разных ЯП приходится выбирать подходящую реализацию интерфейса из множества имеющихся в разных средах [10].

Отметим особенности сред, влияющих на реализацию интерфейса.

Вначале рассмотрим некоторые особенности систем программирования для совре-менных ЯП:

– разные двоичные представления результатов компиляторов для одного и того же ЯП, реализованных на разных архитектурах компьютеров;

– двунаправленность связей между ЯП и их зависимость от среды и платформы;

– параметры вызовов объектов отображаются в операции методов;

– связь с разными ЯП через ссылки на указатели в компиляторах;

– связь модулей в ЯП осуществляется через интерфейсы каждой пары из множества языков (L_1, …, L_n) промежуточной среды.

Современные наиболее распространенные среды (CORBA, COM, JAVA), каждая по своему решает проблему связи разноязыковых компонентов с помощью интерфейса.

Связь компонентов в среде CORBA. В системе CORBA механизм связи разнородных объектов напоминает проектные решения в системе АПРОП с помощью модуля-посредника (stub, skeleton). Модуль-посредник stub выполняет аналогичные функции, связанные с пре-образованием типов данных клиентских компонентов в типы данных серверных компонен-тов посредством

– отображения запросов клиента в операции языка IDL (Interface Definition Language), RMI (Remote Invocation Interface) или API (Application Program Interface);

– преобразования операций IDL в конструкции ЯП и передачи их серверу средствами брокера ORB, реализующего stub в типы данных клиента.

Так как ЯП (C++, JAVA, Smalltalk, Visual C++, COBOL, Ada-95) реализованы на разных платформах и в разных средах и двоичное представление объектов зависит от конкретной аппаратной платформы, в системе CORBA реализован *общий механизм связи* разнородных готовых объектов – брокер ORB (рис. 3).

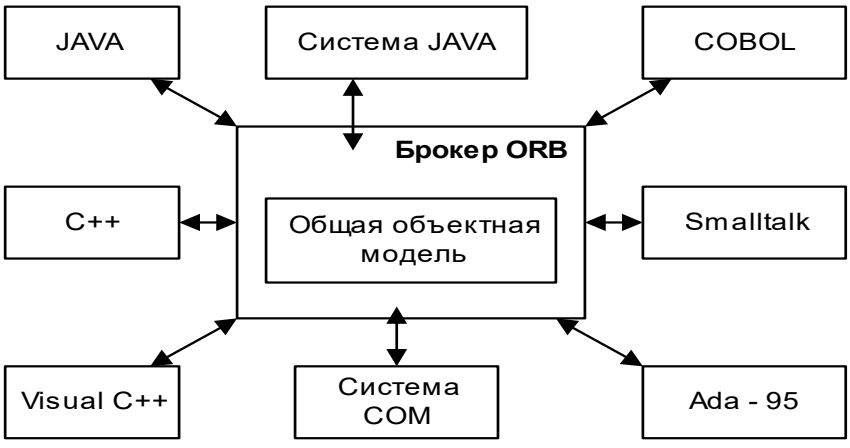


РИС. 3. Интегрированная среда системы CORBA

В эту среду может входить модель COM, в которой типы данных определяются статически, а конструирование сложных типов данных осуществляется для массивов

и записей. В системе CORBA методы объектов используются в двоичном коде, т.е. допускается двоичная совместимость машинных кодов объектов, созданных в разных средах, а также в разных ЯП за счет отделения интерфейсов объектов от их реализаций.

В случае вхождения в состав модели CORBA объектной модели JAVA/RMI, вызов удаленного метода объекта осуществляется ссылками на объекты, задаваемые указателями на адреса памяти. Интерфейс как объектный тип реализуется классами и предоставляет удаленный доступ к нему сервера. Компилятор JAVA создает байт-код, который интересе-тируется виртуальной машиной, обеспечивающей переносимость байт-кодов и однородность представления данных на всех платформах среды CORBA.

В среде клиент-сервера CORBA реализуется два способа связи:

– на уровне ЯП через интерфейсы прикладного программирования;

– на уровне компиляторов IDL, генерирующих клиентские и серверные интерфейсные посредники – stub, skeleton.

Интерфейсы определяются в IDL или APL для объекта-клиента и объекта-сервера, имеют отдельную реализацию и доступны разноязыковым программам. Интерфейсы включают описание формальных и фактических параметров программ, их типов и порядок задания операций передачи параметров, результатов при их взаимодействии. Другими словами, такое описание есть не что иное, как спецификация интерфейсного посредника двух разноязыковых программ, которые взаимодействуют друг с другом через механизм вызова, который реализован на разных процессах. В функции интерфейсного посредника (stub) клиента входит:

– подготовка внешних параметров клиента для обращения к сервису сервера;

– посылка параметров серверу и его запуск для получения результата или сведений об ошибках.

Общие функции интерфейсного посредника (skeleton) сервера:

– получение сообщения от клиента, запуск удаленной процедуры, вычисление резуль-тата и подготовка (кодирование или перекодирование) данных в формате клиента;

– возврат результата клиенту через параметры сообщения и уничтожение удаленной процедуры и др.

Таким образом, интерфейсные посредники задают связь между клиентом и сервером (*stub* – для клиента и *skeleton* – для сервера).

Современные подходы к реализации интерфейса. Кроме перечисленных подходов к решению проблемы интерфейса имеются также следующие:

1) связь готовых, разнородных элементов с помощью интерфейса в IDL, в котором определены входные и выходные данные взаимодействующих элементов;

2) специальный программный интерфейс – JNI (Java Native Interface), допускающий обращение из Java-классов к функциям и библиотекам на других ЯП путем поиска прото-типов обращений к функциям на C/C++, генерации заглавных файлов компилятором C/C++ и обращения из Java-классов к COM-компонентам;

3) технология Bridge2Java, по которой генерируется оболочка для COM-компонента в виде прокси-класса и обеспечивается необходимое преобразование данных для разных ЯП средствами стандартной библиотеки преобразований типов;

4) связь с помощью языка CLR (Common Language Runtime) платформы .Net для любых ЯП, в который транслируются объекты в ЯП (C#, Visual Basic, C++, Jscript) с использованием библиотеки стандартных классов и средств генерации в представление .Net-компонентов;

5) стандартное решение ISO/IEC 11404–1996 спецификации типов данных независимо от ЯП с помощью языка LI (Language Independent) для разноязыковых компонентов, содержащего все существующие типы данных ЯП либо средства их конструирования. В языке LI описываются параметры вызова, как элементы интерфейса, они преобразуются

в типы данных конкретных ЯП специальными правилами и операциями агрегации, представленными в стандарте;

6) XDL-стандарт описания структур данных произвольной сложности и преобразования форматов данных, передаваемых с одной платформы компьютера на другую с помощью специальных процедур;

7) XML-стандарт обеспечения взаимосвязей с преобразованием типов данных ЯП к единому формату XML, понятному многим распределенным средам.

Решения по конкретному преобразованию данных этим не исчерпываются, они еще будут появляться при внедрении новых платформ компьютеров и сред.

Новое толкование интерфейса объектов дал известный специалист в информатике П. Вагнер [17], сформулировав парадигму перехода от алгоритмов вычислений к *взаимодействию объектов*. Суть этой парадигмы: вычисление и взаимодействие объектов – две ортогональные концепции. Взаимодействие – это некоторое действие (action), но не вычисление. Сообщение – не алгоритм, а действие, ответ на которое зависит от операций, влияющих на состояние разделенной памяти (shared state) локальной программы. Он считал операции интерфейса неалгоритмическими, а сообщение – взаимодействием объектов в операционной среде.

Дальнейшее развитие идеи взаимодействия, основанной на действиях, – язык AL (Action language), разработан А.А. Летичевским и Гильбертом [18]. Этот язык определяет вызовы процедур (локальных или распределенных) и их развертку в новую программу в виде операторов действий. Программа из вызовов процедур, т.е. действий рассматривается как ограниченное множество конечных программ, взаимодействующих со средой, в которую они погружаются.

Перспектива повторной реализации системы АПРОП.

Несмотря на приведенные раз-личные подходы к проблеме интерфейса разноязыковых объектов, она по-прежнему остается острой.

Метод сборки специализированных технологий программирования положен в основу метатехнологии, направленный на сборку методов, средств и нотаций в специализированные технологии программирования для конкретной предметной области. Этот подход был впервые сформулирован в 1982г. как технологическая подготовка разработки (ТПР) новых технологий программирования. Этот подход является оригинальным, не имеющий аналогов. Сформулированы задачи ТПР, дано формализованное определение специализированных технологий программирования для решения разных задач (АСУ, СОД, АСНИ и др.) и определен язык спецификаций технологий. К основным элементам ТПР относятся:

обьект разработки (его начальное, промежуточные и конечное состояния);

методы программирования, средства и инструменты, обеспечивающие изменение состояний обьекта;

модели технологических процессов (ТП) и линий (ТЛ);

инженерные методы управления процессом разработки качественных программ по ТЛ.

Для построения новых ТП и ТЛ в рамках ТПР определены классы объектов и язык спецификации ТЛ. Классы обьектов разделяются на понятийные, технологические и инструментальные. В класс понятийных объектов входят модели данных, задач и программ. Все задачи разбиваются на классы по функциональному признаку, каждый из которых определяется на подмножестве схем данных и функциональных элементов (заготовок). В классе типовых задач определяются модели типовых и специализированных программ реализации функций предметной области (ПрО). В класс технологических объектов отнесены модели формализованного представления состояний объектов и процессов их разработки. В их состав входят: система моделей для фиксации проектных решений в ходе разработки; модели процессов и линий для формализации деятельности исполнителей; модель качества программных объектов

для управления качеством разработки на всех этапах жизненного цикла; модель эксплуатационно-программных документов для формирования документации в ходе разработки программного продукта по заданной технологии.

Определены этапы жизненного цикла ПС и дано определение восьми основным этапам жизни отдельных объектов ПС, каждый из которых отображается в одном или нескольких ТП линии. Каждый процесс ТП трактуется как вероятностный автомат, переводящий объект разработки из одного состояния в другое. Заключительным состоянием является готовый программный продукт.

К классу инструментальных объектов относятся методы программирования, средства и инструменты, из которых подбираются более подходящие для создаваемой ТЛ. С их помощью осуществляется целенаправленное преобразование состояний объекта разработки на каждом ТП. Для формального определения входящих в ТЛ процессов разработан язык спецификации ТЛ (ЯСТЛ), который описан в работе «Основы технологической подготовки разработки прикладных программ» (ИК Ан УССР, 1987. -29 с.), являющийся первой попыткой формализованного задания технологий программирования. Основными элементами этого языка являются: технологические процессы, технологические операции, в состав которых входят операции разработки объекта в соответствии с моделью жизненного цикла, документирования и контроля.

Основными атрибутами каждой операции разработки являются: вид состояния объекта, его входные и выходные данные, метод и средство разработки. Описание контрольной операции включает: наименование показателя качества, контролируемый оценочный элемент, метрика и метод оценки. Операция документирования включает указание вида модели документа и набор точек для внесения соответствующих текстов в формируемую документацию.

Данный метод апробирован при создании конкретных функционально-ориентированных технологий программирования прикладных программ, работающих с базами данных. В частности, в рамках АИС "Юпитер" Минобороны СССР разработаны шесть ТЛ, с помощью которых созданы сотни прикладных программ.

Таким образом, результатом разработки метода создания ТЛ являются системы понятий, классов объектов и программная поддержка, включающая ПТК, КОНТИ, ТМНАД, АПФОРС и др.

Литература

1. Глушков В.М., Лаврищева Е.М., Стогной А.А. Моренцов Е.И. – Система автоматизации производства программ - АПРОП, Киев, ИК АН УССР, 1976. -136с.

2. Лаврищева Е.М., Хороле Д.С. Технологические аспекты создания ППП //Общесистемное обеспечение банков данных сетей и комплексов ЭВМ. - Киев. Ин-т кибернетики им. В.М.Глушкова АН УССР. -1986.- С.76-81.

3. Лаврищева Е.М. Принципы технологической подготовки разработки ППО СОД // Проектирование и разработка пакетов программ . - Киев: Ин-т Кибернетики им. В.М.Глушкова АН УССР.- 1987.- С. 34-40.

4. Лаврищева Е.М. Основы технологической подготовки разработки программ СОД.- Киев: 1987.- 29 с. (Препр. 87-5. - Инс-т кибернетики им.В.М.Глушкова АН УССР).

5. Лаврищева Е.М. Технологическая подготовка и инженерия разработки программных средств // Программное обеспечение ЭВМ. Индустрия программного обеспечения. - Тез. докл. Медина. конф.-Калинин: 1987.- Секция 4 ч. 1 .- С. 50-53.

6. Лаврищева Е.М. Технологическая подготовка и программная инженерия // УСиМ.- 1988.- №1 .- С. 48-52.

7. Лаврищева Е.М. Модель процесса разработки программных средств // Там же. 1988.- 5.-С. 43-46.

8. Лаврищева Е.М. и др. Комплекс программных средств, обеспечивающих построение ППП на основе формализованных спецификаций модулей - АПФОРС / Деп.в ЕрНУЦ СНПО "Алгоритм".- 1985.- Инв. 104 и РФАП АН УССР.- 1987.- Инв. АД0002.

9. Лаврищева Е.М., Коваль Г.И., Коротун Т.М., Моренцов Е.И. Программно-технологический комплекс ведения разработки ППО. Технологические документы.- Киев: 1988.-571с. Деп.в РФАП АН УССР .-Инв. АП0218-И.

10. Коваль Г.И., Коротун Т.М., Лаврищева Е.М. Программирование в системе виртуальных машин ЕС ЭВМ.- М.: Финансы и статистика.- 1990.- 254с.

11. Лаврищева Е.М., Грищенко В.М. Сборочное программирование .-Киев.- Наук. думка.- 1991. -213с.

12. Лаврищева Е.М., Грищенко В.Н. Технология сборочного программирования. Основы индустрии программ.- Наук.Думка.-2009.-431с..

По инициативе В.М. Глушкова развиваалась технология программирования и автоматизированные системы – АС, АСУ на государственном уровне СССР. Были созданы

– республиканских и Государственных фондов алгоритмов и программ для складирования, распространения и продажи значимых программ (бумажная документация и носители – магнитные ленты, перфокарты) по организациям страны;

– программостроительного завода (г. Калинин) для сборки новых АСУ из готовых программ и заготовок.

Был создан НИОКР при ГКВТИ СССР по созданию инструментальных систем автоматизации программных комплексов и пакетов прикладных программ (АПРОП, ПРИЗ, ЯУЗА, СИГМАСТАТ, МАКРОБОЛ и др.).

Программы Фондов получили статус изделий или продуктов производственно-технического назначения в 80-е годы и практически использовались по прямому назначению. Их применение во вновь создаваемой системе было затруднено из-за их бумажного представления, отсутствия формализованных интерфейсов с другими програм-мами, необходимости их доработки под конкретные условия. Фонды программ в стране просуществовали более 10 лет и автоматически ушли со сцены вместе с ЕС ЭВМ.

Завод в Калинине проработал несколько лет. На нем были построены два опытных образца АСУ путем модернизации ранее сделанной системы. К тому времени в Фондах заготовок программ для АСУ было мало. Завод, как говорят сегодня, обанкротился из-за отсутствия готовых «деталей» на складе типовых изделий для АСУ, технологии стыковки готовых программ, а также необходимых ресурсов (оборудования, памяти, специалистов, финансов и др.).

В период создания трансляторов в стране начали разрабатываться средства автоматизации программ, в разных ЯП. Одной из таких систем была АПРОП, созданная на основе концепции Глушкова В.М. сборки программных элементов (модулей) по типу сборочного конвейера на фабрике Форда. Об этом он сказал на Ученом Совете в ИК АН СССР 5 марта 1975года был заключен Глушковым В.М. договор с НИЦЕВТ на создание системы автоматизации программ – АПРОП (отв доектор тех. Наук Райков Л.Д.)..

1. Построение системы автоматизации программ и сборки модулей в АСУ и прикладные системы (1975 г.).

Технология АС, АСУ и АСУТП. Теория построения АСУ была изложена Глушковым в работах [1–3], которыми пользуются многие специалисты и сегодня. По его методологии создавались АСУ и ИС (например, на Львовском телевизионном заводе, металлургическом комбинате ГДР и др.). В.М.Глушков предложил систему ОГАС для объединения разных АС и АСУ в единую систему, доступную всем организациям и предприятиям страны. Этот проект не был поддержан ЦК КПСС, так как требовал огромных финансовые ресурсов. Теория АСУ Глушкова постоянно развивается учеными и практиками. В качестве примера можно привести работу аспирантки ИК Украины Н. Т.Задорожной (2004), которая не только использовала идеи и принципы Глушкова, но и практически реализовала ИС документооборота в Академии педагогических наук для сферы образования Украины. Был создан первый учебник по менеджменту документооборота ИС (http://lib.iitta.gov.ua/view/creators), который пользуетсяспросом для управления научными проектами в образовательной сфере Украине [34, 35].

Технология программирования. Методы автоматизации процессов создания системного программного обеспечения новых ЭВМ позволили разработать системы программирования с ЯП (Algol, PL/1, Cobol, Fortran, Modula-2 и др.). Их разработкой, включая адресный язык и ЯП для отечественных ЭВМ, руководила Е. Л. Ющенко. Был создан СМ-метод анализа ЯП [23] в рамках реализации ЯП АЛГОЛ и Кобол для комплекса УВК "Днепр- 2".

В. М. Глушков 5 марта 1975г. на научном семинаре специалистов Института кибернетики выдвинул идею сборки разнородных модулей и программ средствами фабрик, работающих по принципу сборочного конвейера в автомобильной промышленности. Эта идея многие годы плодотворно развивалась в Институте кибернетики по разным направлениям автоматизации программных систем и пакетов прикладных программ (ППП).

Развитие направлений ТП сформулированы В. М. Глушковым в 1980 [28]:

1) модульная система автоматизации производства программ (АПРОП) из стандартизированных программных заготовок в сложные системы [26, 27];

2) метод формализованных технических заданий для проектирования сложных программных комплексов с использованием нескольких алгоритмических языков путем последовательной детализации компьютерного проекта [29];

3) Р-технология программирования систем средствами графического Р-языка, близкого структурному проектированию программ и данных в АСУ ВПК [30].

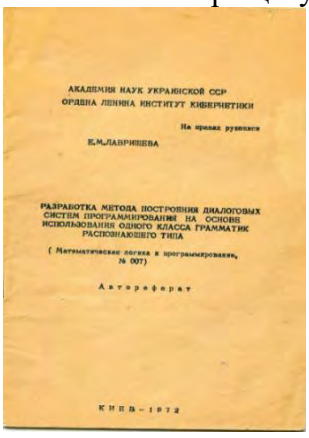
В результате выполнения этих направлений были разработаны: система на основе формализованных технических заданий (Ю. В. Капитонова, А. А. Летичевский); система автоматизации программ – АПРОП (Е. М. Лаврищева); комплексирование пакетов прикладных программ (ППП) для методов численного анализа (И. Н. Молчанов); блочно-сборочное создание ППП статистики и оптимизации (И. Н.Парасюк, И. В.Сергиенко); генерация систем обработки данных из макросов Макробол (Бабенко Л. П.), расширяемая система ТЕРЕМ общих компонентов трансляторов с ЯП (Н. М. Мищенко), система композиции функций ДЕФИПС (В. Н. Редько); технологический комплекс РТК (И. В. Вельбицкий) и др. Данные системы относятся к классу CASE-систем. Они внесли существенный вклад в развитие идеи сборки сложных программ, ППП из готовых модулей и послужили полигоном формирования сборочного программирования программных продуктов на ЕС ЭВМ.

Система автоматизации программ АПРОП обеспечивает: :

- формальное описание модулей и накопление их в Банка модулей;
- задания операций связывания, сборки модулей через данные, передаваемые через модульный посредник, в котором производились преобразования отличающихся типов данных (ТД) связываемых модулей;
- создание Библиотеки интерфесных функций преобразования ТД между модулями с отличающимися данными;
- выполнение операций сборки в операционных средах IBM, Corba, Unix. Intel с использованием Библиотеки модульных интерфесов (БМИ);
- верификация исходного описания модулей и модульных посредников и исправление найденных ошибок:
- трансляция модулей в ЯП на соответствующем трансляторе с ЯП и анализ ошибок для исправления:
- тестирования ПП со сбором данных, оценка надежности и качества систем из модулей:
- сохранение молулей в Банке модулей и интерфейсов (БМИ) для последующего использования

Были защищена кандидатсувя дисс ертация по тематике создания систем программирования АКД и АЛГАМС (см.рис.1) и опубликованы статьи по разработке АПРОП при НИЦЕВИ в 1975г. Главным достижением АПРОП является интерфейс связи отдельных модулей, объектов, устройств и систем, сред на ЭВМ.. **Интерфейс** в системе АПРОП задает связь между всеми видами элементов ВТ и прикладных систем. Начался период создания автоматизированных систем и АСУ ТП. Система АПРОП использовалась в ГДР по договор ГДР ВМНV- СССР МНИИПА (Липаев В.В.) о внедрении УВК «Днепр-2» и создании АСУ ТП в металургии. В.М. Глушков приезжал постоянно в Берлин и учил специалистов этим задачам и руководил созданием АСУ ТП При создании АСУ использовались средства автоматизации ТП, которые представлены в системе АПРОП и Системы программирования АКД и АЛГАМС.

При создании АСУ ТП металлургическими процессами была сделана АСУ ТП медицинскими приборами. За сделанные по договору работы правительства наградила академика В.М.Глушковы группу из 10 человек орденами и медалями. и ТП медицинских приборами использовались системы программирования АКЛ (типа Ассемблер) и АЛГАМС (Вариант АЛГОЛ-60 без рекурсий). В.М.Глушкова. Правительство ГДР наградило нас орденами, медалями, грамотами и знаком АПРОП Лаврищеву Е.М..


АКАДЕМИЯ НАУК УКРАИНСКОЙ ССР
ОРДЕНА ЛЕНИНА ИНСТИТУТ КИБЕРНЕТИКИ
На правах рукописи
Е.М.ЛАВРИЩЕВА
РАЗРАБОТКА МЕТОДА ПОСТРОЕНИЯ ДИАЛОГОВЫХ СИСТЕМ ПРОГРАММИРОВАНИЯ НА ОСНОВЕ ИСПОЛЬЗОВАНИЯ ОДНОГО КЛАССА ГРАММАТИК РАСПОЗНАЮЩЕГО ТИПА
(Математическая логика и программирование, № 007)
Автореферат
КИЕВ - 1972


URKUNDE
Kolln. Katja Lawrischtschewa
IST MITGLIED DES KOLLEKTIVS
DNEPR 21, des Instituts für Kybernetik der Akademie der Wissenschaften der USSR
DAS FÜR HERVORRAGENDE LEISTUNGEN BEIM AUFBAU DES SOZIALISMUS UND BEI DER FESTIGUNG UND STÄRKUNG DER DEUTSCHEN DEMOKRATISCHEN REPUBLIK
MIT DEM EHRENTITEL
KOLLEKTIV
DER SOZIALISTISCHEN ARBEIT
AUSGEZEICHNET WURDE
Berlin 15.6.1971
WERKDIREKTOR

.

.

Были опубликованы статьи и книги:

1. Грищенко В.Н., Лаврищева Е.М. О создании межъязыкового интерфейса для ОС ЕС ЭВМ // УСиМ. 1978. № I.

2 Система автоматизации производства программ (Глушков В.М,, Стогний А.А., Никитин А.И., Лаврищева Е.М. и др.), Киев, 1976.-137с.

3.Защищена кандидатская диссертация Е.М.Лаврищевой в 1972г.

В рамках АСВТ были созданы 4 модели – М-1000, М-1010, М-2000 и М-3000. Модели с помощью интерфейсов объединялись в произвольные комплексы. Это способствовала повышению производительности работ ПО на УВК. АСУ ТП и АСОД с данными, которые накапливались в Базах данных и БМИ. В состав первой проекта Ряд 1 на ЕС ЭВМ, где использовались М 1-2, М1030, М1040 на специализированных ЭВМ М-40, М-60, для решения задач взаимосвязи сложных комплексов. См. работы Б.Н.Наумова, В.В.Рязанова. Малые ЭВМ и их применение.-1980. -М.: 231с.; Тюрин В.Ф. Операционные системы Диспак. -М. Наука. - 1985. -336с.; Глушков В.М. Введение в АСУ. -К. Техника. -1972. -310с. и др. Первая АСУ ТП сборки медицинских приборов под руководством академика В.М.Глушкова была создана в ГДР в 1977 году и использовалась в Болгарии, Венгрии и др.

Развитие работ по системе автоматизации программ на ВПК в 1978-1983гг

В.М. Глушков выступил в МинРадиопром с докладом по автоматизации приборов и прикладных программ средствами системы АПРОП. Был заключен договор с МНИИПА Министерства Радиопромышленности СССР под руководством Липаева В.В.* на разработку средствами системы АПРОП технических приборов для авиации, космонавтики и морфлота. Были изготовлены комплексы программного обеспечения ЯУЗА, ПРОТВА, ПРОМЕТЕЙ и РУЗА и набор радиотехнических и бортовых средств для авиации, космоса и флота. При реализации задач ВПК была создана книга «Связь разноязыковых модулей в ОС ЕС», Лаврищева Е.М., Грищенко В.Н. и переданы в ГОСФАП для использования в 50 организациях страны, а также базовые средства комплексов ПРОМЕТЕЙ в ЕрНУЦ 1984году и сданы в ЕрНУЦ.. За участие в этих работах основные разработчики проекта ВПК Липаева В.В. были награждены Государственной премией Кабинета Министров СССР (1985), включая автора Лаврищеву Е.М. В.М. Глушков умер в 20января 1982. .

1.2.1. Сущность метода сборки разнородных программных и технических элементов для создания системного сборщика в Интернет

В.М. Глушков после конгресса IFIP в Мюнхене в 1972г. поставил перед Ученым советам задачи конвейерной сборки (по Форду в автомобильной промышленности, где детали соединялись через болты и гайки) прикладных систем представления знаний. С этого момента ЛЕМ включилась в реализацию этой идеи при программирования задач предметных областей знаний.

Был создан метод сборки модулей, который основывался на интерфейсном посреднике, в задачу которого входила эквивалентное преобразование передаваемых и возвращаемых данных. Разработана теория представления абстрактных типов данных ЯП и подходов к формальному их преобразованию в связываемых программных элементах (модулей, компонентов), различающихся форматами представления их трансляторами к форме представления в памяти ЭВМ.

Модули, интерфейсный связник-посредник, библиотека интерфейсных функций (БМИ) и операции сборки – основа сборочного программирования, реализованная в автоматизированной системе АПРОП и книге Связь разноязыковых модулей. В первой публикации по системе АПРОП и методу сборки модулей отработана концепция В.М.Глушкова на конвейерную сборку программ на машинах ЕС, автоматизирующих связи разноязычных модулей (1976).

Метод сборки задавался операторами Link <имя модуля>&<имя модуля>. На основе такого оператора генерировался модуль-связник, в функции которого входило отображение фактических параметров модуля в типы другого модуля, проверка соответствия параметров (количество и порядок), форматов данных в ЯП и в памяти машины и др. Сгенерированный модуль-посредник содержал обращения к вызываемым элементам библиотеки интерфейса, которые выполняются в момент перехода связника от одного модуля к другому и обратно.

Метод сборки реализовывался процессами:

1) обработка паспортных данных модулей;

2) анализ операторов Call и Link и составление задания на их обработку;

3) генерация модуля-посредника, составление таблицы матрицы соответствия пар компонентов и преобразование ТД связанных модулей (b-boolean, c-character, i-integer, r-real, a-array, z-record и др.) через обращение к функциям библиотеки интерфейса (БМИ);

4) интеграция пар модулей и их размещение в базе данных системы для всех пар в сложную структуру ПС;

5) трансляция и компиляция модулей собранной структуры к виду готовой программной структуры;

6) трассировка интерфейсов и отладка функций модулей в каждой паре структуры;

7) тестирование структуры в целом;

8) формирование готового программного продукта (ПП) и проверка на качество для запуска и управления инсталляцией и выполнением.

Система АПРОП стала первой системой автоматизации взаимосвязи разноязычных модулей с помощью интерфейсных посредников для языков четвертого поколения (stub, skeleton в Corba) и в проектах Липаева В.В. Руза, Яуза и Прометей.

Разработанный метод сборки и программные средства его поддержки стали базисом автоматизированной инженерии программирования модулями в системе АПРОП. В публикации по методу сборки модулей были приведены основные пути реализации индустрии разноязыковых программ на машинах ЕС (1976) [24].

Главные объекты системы АПРОП: модуль, модуль-посредник, межмодульный интерфейс, Банк модулей и метод проектирования систем снизу–вверх с выбором готовых модулей из Банка модулей и их сборки в новые программные структуры. Базовая концепция системы – интерфейс как аппарат связи ЯП и модулей, записанных в разных ЯП (Фортран, ПЛ/1, Алгол-60, Ассемблер, Кобол). Каждый исходный интегрируемый ("погружаемый") в АПРОП прикладной модуль имел паспорт, в котором описывались сведения о назначении, объеме, параметрах и др. Среда была одна для всех – ОС ЕС (IBM-360).

Сборка готовых модулей базируется на совокупности модулей, их паспортах и операторах сборки отдельных функциональных элементов:

1) оператор сборки Link, задающий сборку двух разноязычных объектов или модулей графа;

2) Link seg A (A2, A3, *A4) – связать модули A, A3 и A4 в сегментную структуру A, где A4 вызывается динамически;

3) Link Prog B ((B1, B2), C= X(C1), D=(Y, D1=Y1)) – объединить модули B1, B2, к ним присоединить C и D с параметрами C1. Y, D1.

4) оператор //EXEC – выполнить *модуль* A_1 //PL *Trans* A_1 .

5) генерировать интерфейсный связник mod-interface generation for $A_1 \cap A_2$ и др.

Правила сборки определяют совместимость объединяемых объектов, которые содержат описание функций для согласования разных характеристик, представленных в их паспортах.

Процесс сборки объектов может проводиться ручным, автоматизированным и автоматическим способами. Как правило, последний способ невозможен, связан с недостаточно формальным определением программных КПИ и их интерфейсов. Ручной способ нецелесообразен, так как сборка готовых КПИ представляет собой большой объем действий, которые носят скорее рутинный, чем творческий характер. Наиболее приемлемый способ – это автоматизированная сборка, когда по заданным спецификациям программ осуществляется сборка с помощью стандартных правил сборки разнородных объектов.

Средства, которые поддерживают данный способ сборки, называют инструментальным средствами сборочного программирования. К ним относятся средства комплексирования (объединения компонентов в более сложный объект); интерфейсные средства описания и использования моделей программирования (совокупность моделей интеграции разных программных объектов).

Необходимыми условиями применения этого метода программирования является:

1) наличие большого количества разнообразных КПИ, как объектов сборки;

2) паспортизация объектов сборки;

3) наличие довольно полного набора стандартных правил сопряжения объектов, алгоритмов их реализации и средств автоматизации процесса сборки;

4) технологии линией с последовательностью операций постепенного изготовления и установления связей между КПИ при образовании системы или семейства ПП.

Последнее условие означает, что должны существовать определенные формы представления ПС как знаний о предметных областях, универсальные с точки зрения проектирования и разработки ПС. Основное задание сборки – выявление типов связи, описание их в интерфейсе и реализация в виде посредника интерфейсов между отдельными модулями и/или компонентами, который обеспечивают их "стыковку" или связь в процессе выполнении в некоторой среде.

Развитие метода сборки. Идея сборки с помощью интерфейса развивалась за рубежом в проектах MIL, SAA, IBM, Sun, Oberon, CORBA и др. В настоящее время идея сборки стала типовой в классе традиционных и современных ЯП. Она реализована аналогично в названных системах и базируется на теории преобразования нерелевантных типов данных в ЯП и описана в руководствах по применению. Новые подходы к сборке опубликованы в ряде работ, в том числе у И. Бея («Взаимодействие разноязычных программ», 2005) и др. Система CORBA для объектных элементов реализовала универсальный управляющий связник – брокер объектных запросов, который связывает готовые объекты-методы и компоненты в любых ЯП через посредники – stub (туда) и skeleton (обратно). Интерфейсный посредник объектов описывается в новом языке IDL (Interface Definition Language). В нем параметры передачи данных помечаются *in* и *out* между разнородными объектами в любых ЯП. Данные готовых объектов и КПИ содержат описание *типов данных* в соответствующем ЯП и при их передаче от одного объекта к другому они проверяются на соответствие описания в посреднике stub, их релевантность параметров из skeleton.

При преподавании в МФТИ с 1993 на кафедре В.М.Глушкова создан учебник в 2006г.:

«Методы и средства инженерии программного обеспечения» Лаврищева Е.М., Петрухин В.П., изданный после конкурса в МГУ «Программная инженерии» представлен на сайте университета

Интернет intuit.ru с 2007 и им пользуются студенты и получили дипломы более 989 студентов. (В 2016 Петрухин В.А. издал 2 версию этого учебника).

Издана книга Лаврищевой Е.М. Методы программирования. Теория, методы, практика. 2006.- 471с. Наук.думка. В ней рассмотрены действующие на этот момент зарубежные и отечественные методы программирования. См. раздел Библиотека: книги, препринты и учебники.

Развитие сборочного программирования для трансляторов с ЯП в ИК АН

В отделе Глушкова В.М. (1975--1990) после выступления на ученом Совете ИК АН Украины о поездки на IFIP по вопросу конвейерной сборке в автомобильной промышленности фабрики Форда, он сказал, что я вижу, что и компьютерные программы будут собираться в сложные структуры из готовых элементарных подпрограмм и модулей. В его отделе начались исследования этой идеи в момент, когда начали создаваться трансляторы с ЯП. Первым специалистом откликнулась на эту идею старший научный сотрудник Мищенко Надежда Михайловна (МНМ), которая с этой идеей сборочного программирования трансляторов выступала на многих Всесоюзных конференциях. Так на конференции 198г. после смерти В.М.Глушкова многие делали доклады по разным напралениям научных идей, выступали Лаврищева Е.М. и Мищенко Н.М.

На Всесоюзной конференции в Таллине «Автоматизация трансляторов и ПП» Таллин, 1982г.- Институт Кибернетики ЭССР, Тыугу Э.Х. Программный комитет: Тыугу Э.Х., Ершов А.П., Лавров С.С., Редько В.Н., И.В. Поттосин.

Пленарные доклады:

1. Ершов А. П. Фундаментальные процессы трансляции.
2. Лавров С. С. Язык ДЕКАРТ.
3. Курочкин В. А., Серебряков В. А. Современные методы описания языков.
4. Бежанова М. М., Тыугу Э. Х. Пути построения пакетов программ.
5. Вооглайд А. О., Мериете М. В. Обзор систем построения трансляторов.

Секции конференции:

Секция 1. Технология трансляторов. Председатель А. П. Ершов, секретарь М. В. Мериете.

- Мищенко Н. М. Определение семантики входного языка расширяющейся системы программирования ТЕРЕМ.
- Щеголева Н. Н. О погружении языков программирования и проектирования в вычислительную среду системы ПРОЕКТ.
- Бублик В. В., Гороховский С. С., Чуйкевич В. С. Методы определения языков программирования для систем интерпретирующего типа.

Секция 2. Методы трансляции. Председатель С. С. Лавров, секретарь М. Томбак.

• Федюрко В. В., Фелижанко О. Д. О методах реализации специализированных языков управления процессами функционирования системы программ.

Секция 3. Теория программирования. Председатель И. В. Поттосин, секретарь Виллемс.

Секция 4. Построение пакетов программ. Председатель В.М.Курочкин, секретарь Д. Лий.

• Лаврищева Е. М. Транслятор с языка Д-АЛГАМС. Подход к автоматизации пакетов прикладных программ.

Секция 5. Реализованные СПТ. Председатель В. Н. Редько, секретарь Х. Р.

На этой конференции с докладом выступила ЛЕМ по Трансляторам АКД и Д-Алгамс, вызвала интерес и опубликован в Сборнике этой конференции. Позже была конференция по технологии программирования. Там же. В 1984 Эн Харальдович Тыугу предложил быть первым оппонентом докторской диссертации Лаврищевой Е.М. «Методы, средства и инструменты сборочного программирования СОД» (ВАК СССР, 1989).

На Всесоюзной конференции 1982 г. были представлены доклады Мищенко Н.М «Определение семантики входного языка расширяющейся системы программирования ТЕРЕМ», в задачу которой входило создание методом сборки трансляторов с ЯП, которые представлялись на других конференциях и опубликованы в журналах ИК АН УССР и УСиМ и др.

Н.М. Мищенко "Об адаптации программных модулей, предназначенных для сборки языковых процессоров". Сборник конференции ВПК, 1982. - С.59-70.

Аннотация. Актуальность задачи автоматизации разработки языковых процессоров подтверждается не только появлением новых языков программирования, но и необходимостью адаптировать уже существующие языки к конкретному применению. Свойство расширяемости языков и соответствующих систем программирования является предпосылкой для адаптации реализованных систем программирования к использованию в новых условиях. В РСП ТЕРЕМ разработаны следующие средства адаптации:

1. Автоматическое порождение синтаксических таблиц входного языка;
2. Адаптация как процесс развития входного языка, в роли которого предусматривается класс языков, описываемых модифицированными БНФ;

Щоголевой Н.М., Валькевич Т.А. и Мищенко.М. «Средства ПО транспьютерной системы, включающей общесистемное ОМО, языковые процессоры, системы программирования многопроцессорных комплексов».

Использован принцип сборочного программирования для языковых процессоров из совокупности модулей системы:

1. Система сборочного программирования (ССП):

– комплекс программных модулей (РСП ТЕРЕМ) на ПК Мир (Мищенко, Щоголева);

– словарный модуль на ЕС 1766 (Валькевич).

2. Архитектура системы программирования ССП. Принцип декомпозиции мультимодульных программ (Мищенко).

3.Методика использования ССП (Мищенко) для транспьютерного использования в Макроконвейр системы Терем и на ПК.

Конференции в Днепропетровске 19-20 сентября «ОМО и системное программирование.

Мищенко Н.М. "*Об одном комплексе программных модулей для сборки трансляторов.*

Аннотация. Рассматриваемый комплекс предназначен для реализации однопроходных трансляторов на ЕС ЭВМ для класса языков, который включает языки программирования и подмножества естественных языков, используемых для общения в человеко-машинных системах и описываемых нелеворекурсивными контекстно-свободными грамматиками.

В программных модулях комплекса реализованы хорошо изученные и наиболее часто встречающиеся в трансляторах функции перевода и структуры данных. Предполагается участие

пользователя в создании непредусмотренных в комплексе модулей или "своих" версий уже имеющихся модулей по прилагаемой к комплексу методики сборки.

Статья в журнале УСиМ

Н.М. Мищенко "*Средства расширения входных языков РСП ТЕРЕМ и их применение*"// УСиМ. – 1990. – № 5. – С. 55-62.

Аннотация. В статье рассмотрены языковые и программные средства одноуровневых и многоуровневых расширений контекстно-свободных языков, входящие в состав РСП ТЕРЕМ и наследуемые каждым языковым процессором, реализуемым на ее базе. Система допускает также класс исходных базисных языков, описываемых наборами ключевых слов.

В Сборнике ИК статья «Мищенко Н.М. О сборочном программировании языковых процессоров / Сб. Интеллектуализация программного обеспечения информационно-вычислительных систем. Сб.науч. тр. – Киев: Ин-т кибернетики им. В.М. Глушкова АН УССР, 1990. – С. 45-52.

Аннотация. В работе рассматривается способ реализации языковых процессоров путем их сборки из готовых программных модулей, которые вместе со средствами и методикой их адаптации к конкретному реализуемому языку образуют систему сборочного программирования языковых процессоров.

Статья в журнале УСиМ

Н.М. Мищенко "Средства расширения входных языков РСП ТЕРЕМ и их применение"// УСиМ. – 1990. – № 5. – С. 55-62.

Аннотация. В статье рассмотрены языковые и программные средства одноуровневых и многоуровневых расширений контекстно-свободных языков, входящие в состав РСП ТЕРЕМ и наследуемые каждым языковым процессором, реализуемым на ее базе. Система допускает также класс исходных базисных языков, описываемых наборами ключевых слов.

В Сборнике ИК статья «Мищенко Н.М. О сборочном программировании языковых процессоров / Сб. Интеллектуализация программного обеспечения информационно-вычислительных систем. Сб.науч. тр. – Киев: Ин-т кибернетики им. В.М. Глушкова АН УССР, 1990. – С. 45-52.

Аннотация. В работе рассматривается способ реализации языковых процессоров путем их сборки из готовых программных модулей, которые вместе со средствами и методикой их адаптации к конкретному реализуемому языку образуют систему сборочного программирования языковых процессоров. Основные результаты работ Мищенко Н.М. по сборке трансляторов в период 1970-80-х годов было:

- построение инструментальной системы программирования (РСП) ТЕРЕМ и использование ее для построения трансляторов Макроконвейєрного типа с языков, грамматика которых описывались в БНФ.

РСП ТЕРЕМ – это универсальная синтаксическая подсистема Анализатора входных языков в синтаксической таблице и универсальная программа семантической подсистемы, открытой для расширения. Транслятор, который разрабатывался для языка с допустимого класса языков, проверялся Анализатором, а по БНФ-описанию граматики языка программ Конструктор строил синтаксические таблицы, которые с Анализатором образовывали синтаксическую подсистему транслятора. В состав семантической подсистемы входили способы систем построения трансляторов (СПТ). Их наличие в составе новопостроенной системы программирования, позволяющей специалисту изменять семантику входного языка в процессы использованиия транслятора. Отсутствие способов СПТ в трансляторе означало, что входной язык фиксирован и не требует развития в час использования.

Есть третий вариант включения способов СПТ в транслятор для развития входного языка, после чего способы СПТ могут быть выключены из транслятора. Оригинальной была связь синтаксической и семантической подсистем: к элементам описания синтаксиса в БНФ дописывались имена соответствующих семантических действий со знаками действий, которые регулируют час инициации действий с фразой, опознанной в час синтаксического анализа.

Способность входного языка к изменениям, необходима для реализации новых языков, которые эволюционизируют в процессы их использования. Процессы и их использование, описаны средства РПС ТЕРЕМ и описываются свойства трансляторов, построенных в системе РСП ТЕРЕМ, в которой реализованы:

- исследовательский характер реализации языка, если в его трансляторе включены средства СПТ;
- адаптивность системы до конкретних условий вычислительной системы;
- практичность системы, поскольку 3/5 нового транслятора – это средства РСП ТЕРЕМ.

Сделаны два транслятора средствами РСП ТЕРЕМ с языков ПЛ/1и Си, для работы на персональных компьютерах (ПК). На первом ПК Мир 1 производилась трансляция программ в языке Си и использовалась на Мир 2.

Мищенко Н.М. подготовила докторскую диссертацию по технологии сборочного программирования трансляторов с ЯП (PL/1, Fortpan, Си, Snobol и др.). Но защита не состоялась из-за требований Ученого Совета к защите докторской диссертации, которые она не успела выполнить из-за болезни.

1 2.1. Развитие сборочного программирования в ИСИ Новосибирске

Развитием метода сборки руководил Ершов А.П. в ИСИ. Им , его сотрудники и др. издали работы:.

- Ершов А.П. Введение в теоретическое программирование. -1977. Из.-во Наука.-280с.
- Лаврищева Е.М., Грищенко В.Н. Связь разноязыковых модулей в ОС ЕС. -1982. - М. 137с.
- Касьянов В.И. Применении теории графов для построения системного ПО. - 1975-1982. – Новоссибирск.-37с.
- В.А Евстигнеева –Применение теории графов в системном программировании» 1985. – Новосибирск, 151с.
- Липаев В.В., Позин Б.А. Штрик. Технология сборочного программирования. - 1992.-Москва.-281с.
- Лаврищева Е.М., Грищенко В.Н. Сбороное программирование. Основы индустрии программных продуктов. –Академпериодика.- 2009. – 431с.
- В.П.Котляров Использование сборочного программирования в образовании 1998.-2021. -ИСП РАН конференция.
- А.Недоря Техническая сборки для промышленности 1991.-Доклад на эту тему Алексей Недоря (алексейнедоря@rf.ru). Сразу после развала СССР начал работать в Европе и развивал сборочное программирование для технических средств. В 2019 А. Недоря сделал доклад http://алексейнедоря.рф/?p=338 для мини и макро ПК сборочного программирования задач для них. См. доклады ниже:

1. **Алексей Недоря** В 2020 году он делает доклад на эту тему и со средствами проверки на безопасность и защиту данных. Ниже приводится его доклад

Ворчалки о программировании

Алексей Недоря о программировании и не только

- Компоненты

- Разработка языка программирования
- Вир-2

Вир

Стандартизация программирования« Настоящее» программирование Человеко-ориентированный Интернет».

2.Летичевский А.А., Капитонова Ю.В.– Инсерционное программирование (1). Режим совместимости языков. 1979.

3.Статья В.П. Котлярова на Открытой конференции ИСП РАН им. В.П. Иванникова Москва, 5-6 декабря 2019 г.

В рамках процесса совершенствования экосистемы разработки приложений для различных устройств Huawei компания работает над новым языком программирования. В статье рассматривается подход к реализации ООР в языке программирования, который рассматривается как движение в сторону компонентно-ориентированного программирования.

2. http://www.iprinet.kiev.ua/gf/nau_pp_comp.htm

3. Доклады ЛЕМ делались на многих Всесоюзных конференциях в Кишиневе, Киеве, Грузии, Минске, Новосибирске, Ленинграде, Прибалтийские странах и Средней Азии (1978-1989) и печатались в Сборниках конференций.

Открытая конференция ИСП РАН им. ИВАННИКОВА 2018

Доклад

Информатика и ЭВМ-70лет. Аспекты развития

Е.М.Лаврищева
д.ф.-м.н., профессора, гнс. ИСП РАН

Сборка вариантов программных, технических и операционных систем

По приезду 17 июня 2014 в Москву ЛЕМ съездила в МФТИ в Долгопрудный с документом, что преподавала на кафедре Глушкова «Оптимизация автоматизированных систем» в филиале

МФТИ в ИК НАНУ с 1993, руководила 12 магистерскими работами по тематике программной, компьютерной инженерии и технологии программирования (на укр. языке) и защити по театике моделированияи программирования несколько канд. Диссертаций (Грищенко В.Н., Коваль Г.И., Моренцов Е.И., Коротун Т.М., Стеняшин А.И., Колесник А. К., Слабоспитская О.И. и др.). Обучила более 2500 студентов МФТИ и КГУ.

Ректор МФТИ позвонил на кафедру вычислительная математика и информатика член-корр. Петрову И.Б. и предложил взять меня на работу. Сдала заявление и дипломы об образовании. Была зачислена на эту кафедру С 1 сентября 2014 зачислена на должность профессора (1989, ВАК СССР) – читать курс «Программная инженерия и технология программирования» согласно учебника «Методы и средства Инженерии ПО МФТИ в intuit.ru и новой монографии (2014) «Software Engineering компьютерных систем. Технология, парадигмы, CASE-средства программирования», изд.-во Наук. Думка, К.: 2014. -284с.

- Лаврищева Е.М. Компонентная теория и коллекция технологий для разработки индустриальных приложений из готовых ресурсов, Труды 4-Труды 4- Научно-практической конференции «Актуальные проблемы системной и программной инженерии», АПСПИ-2015, 20-21мая 2015, с. 101-119.

Для целей автоматизации ЖЦ разработки ПС в ИСП проводились работы по исследованию средств создания онтологий – в Семантик Веб ждя описания ЖЦ стандарта ISO/IEC 12207 Life Sycle -1998. 2007 и Вычислительной математики. Сделаны статьи по этой тематике:

- Lavrischeva Ekaterina. Ontological Approach to the Formal Specification of the Standard Life Cycle, Conference "Science and Information Conference-2015, July 28-30, London, UK, www.conference.thesai.org.- p.965-972.

- Ekaterina M. Lavrischeva. Moscow Physics-Technical Institute, Dolgoprudnuy, Russia

Ontology of Domains. Ontological Description Software Engineering Domain—The Standard Life Cycle", Journal of Software Engineering and Applications, 2015, 8, 324-338

- The Operating Computing Complex «Dnepr-2», Ekaterina Lavrischeva, Institute of the System Programming of Russian Academy of Sciences, ispras.ru.-SoRuCom-2014, IEEE Springer-2015, с.102-104 (Lavrischeva Dnepr-2 -6.12.2014).

- Лаврищева Е.М., Слабоспитская О.А. Технология моделирования изменяемых программных продуктов и систем //XII Межд. Научно-практ. конф. «Теоретические и прикладные аспекты построения программных систем».-TAAPSD'2015, 23-26 ноября, 2015.-с.118-128.

- Ekaterina M. Lavrischeva. Moscow Institute of Physics and Technology (State University) MIPT, Dolgoprudny, Russia. Assembling Paradigms of Programming in Software Engineering, Journal of Software Engineering and Applications, 2016, 9, 296-317.

Международной конференции АПСПИ-2015

Был сделан доклад и опубликована статья **«Лаврищева Е.М. Компонентная теория и коллекция технологий для разработки индустриальных приложений из готовых ресурсов**, Труды 4-Труды 4- научно-практической конференции «Актуальные проблемы системной и программной инженерии», АПСПИ-2015, 20-21мая 2015, с. 101-119.

Сделан доклад в ИСП про парадигмы сборочного типа (см.выше).

Conference "Science and Information Conference-2015, July 28-30, London, UK.

Был сделан доклад (см.ниже)

Lavrischeva Ekaterina. Ontological Approach to the Formal Specification of the Standard Life Cycle . london lavr6+2021. www.conference.thesai.org.- p.965-972.

На этой конференции представители Международного комитета IEEE предложили сделать патент на тему автоматизации ЖЦ. В ИСП Виктор Петрович принял решение сделать государственную программу Автоматизации ЖЦ (2018).

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2018615442

«Реализация метода онтологического моделирования домена ЖЦ стандарта ISO/IEC 12207»

Правообладатель: ***Федеральное государственное бюджетное учреждение науки Институт системного программирования им. В.П. Иванникова Российской академии наук (RU)***

Автор: ***Лаврищева Екатерина Михайловна (RU)***

Заявка № **2018612585**
Дата поступления **20 марта 2018 г.**
Дата государственной регистрации в Реестре программ для ЭВМ ***08 мая 2018 г.***

Руководитель Федеральной службы по интеллектуальной собственности

Г.П. Ивлиев

Процессы поддержки ЖЦ включают все процессы, которые выполняются после построения модели и проверки ее работоспособности. Онтологическая структура отражает процессов ЖЦ и

их представление в XML-языке средства DSL Tools VS.Net (для трех категорий процессов ЖЦ - описание ЖЦ в языке XML занимает 15 страниц. По онтологии (анг.) были напечатаны статьи:

- Lavrischeva Ekaterina. Ontological Approach to the Formal Specification of the Standard Life Cycle, Conference "Science and Information Conference-2015, July 28-30, London, UK, www.conference.thesai.org.- p.965-972.

- Ekaterina M. Lavrischeva. Moscow Physics-Technical Institute, Dolgoprudnuy, Russia. Ontology of Domains. Ontological Description Software Engineering Domain—The Standard Life Cycle", Journal of Software Engineering and Applications, 2015, 8, 324-338

- The Operating Computing Complex «Dnepr-2», Ekaterina Lavrischeva, Institute of the System Programming of Russian Academy of Sciences, ispras.ru.-SoRuCom-2014, IEEE Springer-2015, c.102-104 (Lavrischeva Dnepr-2 -6.12.2014).

Основным элементом в этом процессе является Интерфейс (модульный, программный, техническихЪ , который был представлен в на конференции «Интерфейс-СЭВ» и пролучено свидетельство ЛЕМ, Коваль Г.И,, Коротун Т.М.

Свидетельство

Свидетельство «О государственной регистрации программы автоматизации ЖЦ 12207, по докладу в Лондоне для ЭВМ № 2018615442. Государственная служба по интеллектуальной собственности от 28.03.2018.

Разработаны методические пособия для обучения студентов МФТИ -www.mipt.ru. (опыт преподавания дисциплины «Программная инженерия» в филиале МФТИ ИК НАНУ проводился с 1992г. на кафедре В.М. Глушкова «Методы оптимизации АСУ МФТИ После переезда в Москву в 2014 преподавание курса «Программная инженерия» продолжился. И зав. кафедрой Петров И.Б. предложил дать новые версию методических пособий. К ним относятся следующие:

- Лаврищева Е.М. Программная инженерия. Тема 1. Теория программирования. Учебно-методическое пособие. Москва, МФТИ, 2016. 48 с.

- Лаврищева Е.М. Программная инженерия. Тема 2. Технология программирования. Учебно-методическое пособие. Москва, МФТИ, 2016. 52 с.

- Лаврищева Е.М. Программная инженерия. Тема 3. Базовые основы программной инженерии. Учебно-методическое пособие. Москва, МФТИ, 2016. 52 с.

См. раздел - Библиотека книг, препринтов и учебников.

Были сделаны доклады на конференции «Научный сервис в сети Интернет», TAAPSD-2015, ИПИ РАН «Моделирования и проектирования информационных систем, РФФИ».- Таганрог и опубликованы.

Доклад–статья Е.М. Лаврищева, Л. Е. Карпов, А. Н. Томилин. Семантические ресурсы для разработки онтологии научной и инженерной предметных областей, Всероссийская конференция "Научный сервис в сети Интернет" 21-26 сентября 2015, Абрау-Дерсу. –с.193-218. Наука и сервис а Интернет 2015.

ПУТИ РАЗВИТИЯ ИНДУСТРИИ ПРОГРАМНОЙ ПРОДУКЦИИ
по Глушкову
2013

Доклад в КНУ
доктора физ.-мат. наук, профессора
Е.М.Лаврищевой

Лекция в МФТИ
2021

Лаврищева Е. М., Грищенко В. Н. Сборка модулей, объектов и компонентов в языках программирования.— М., ИСП РАН.—2022 — 174 с.

Книга посвящена методам сборки модульных элементов, записанных в языках программирования (ЯП) ОС ЕС и интеллектуальных ресурсов в WWW3C. Цель работы – научить программистов методу сборки модулей в разных ЯП для создания сложных программных систем и комплексов путем их конструирования из готовых ресурсов с использованием разработанной библиотеки межъязыкового интерфейса (БМИ). От пользователей требуется знание ЯП высокого уровня для описания функций прикладных систем модульными элементами в современных ЯП в средах IBM 360 (ОС ЕС), VS.Net, Java.Net, Corba и др. Приводятся функции БМИ для преобразования передаваемых простых, сложных, неструктурированных данных от одного ресурса к другому с учетом современных общих типов данных GDT (General Data Types) стандарта ISO/IEC 11404-2012 и метода сборки ресурсов, записанных в современных ЯП (Algol, Cobol, Smalltalk, Fortran, Java, Ruby, Basic, C, C++ и др.) в средах Интернет (IBM, VS.Net, Corba, Grid, Etic, BSD, Oracle BD и др.) для получения систем в медицине, биологии, физических экспериментах, нано приборах и роботах.
(https://www.ispras.ru/lavrishcheva/monographies_lavrishcheva_11.php - Монография по создание СП и СПС из объектов, модулей и сервисов Интернет.

Р а з д е л 3

Подходы к интеллектуализации описаны в разделах отчета за 2016-2017, обсуждались в докладах конференций «XIII Российский Фестиваль науки “NAUKA 0+” 2018 и на Конференции ISPRAS OPEN -18 и отображена в статьях. См. доклады:

- Лаврищева Е.М. Современные системы искусственного интеллект», 13.10.2018. в Финансовом университете при Президенте XIII Всероссийского фестиваля науки “NAUKA 0+”, симпозиума «Искусственный интеллект: различные подходы к его воплощению»;

- Аветисян А.И., Лаврищева Е.М. «Информатика и ЭВМ. Анализ и аспекты развития» Открытая конференция ИСП РАН им. В.П.Иванникова, 22-23.11.2018. сделан доклад. Потом в направлении автоматизации жизненного цикла ПО (ISO/IEC Cycle Life 12207-2007) сделан доклад Лаврищевой “Ontological Approach to the Formal Specification of the Standard Life Cycle, "Science and Information Conference-2015", Jule 28-30, London, UK,www.conference.thesai.org.- p.965-972;

- Lavrischeva E.M. Ontology of Domains. Ontological Description Sofware Engineering Domain - The Standard Life Cycle, Journal of Software Engineering and Applications, July 24, 2016).

На эту статью много запросов в среде Интернет из Китая, Тайланда, Индии и др. Сегодня интеллектуализация касается Больших данных (Big Data). Это направление особенно стало актуальным в связи с роботизацией разных сфер человеческой деятельности, в том числе работы с огромными объемами данных, получаемых с космоса, недр земли, океана и др. (См. статья. Лаврищева Е.М. Рыжов А.Г. Применение теория общих типов данных стандарта ISO/IEC 12207 GDT применительно к Big Data.- The cconference “Actual problems in science and ways their development”, 27 desember 2016, http://euroasia-science.ru/ p.99-110.

Статья, посвящена информатике и концепциям программирования информационных и программных систем ИСП РАН с учетом проекта Информатизации России, Lavrischeva E.M., Petrenko A.K. Informatics. Formation of computer software and technologies of software systems. Trudy ISP RAN/Proc. ISP RAS, 2018.

Далее описыватnся наука моделирования и программирования сложных программных и технических систем предметных областей знаний, изготовляемых по проектам РФФИ 352 и РФФИ 206 2019-2021 в ИСП РАН . с начала появления ЭВМ в нашей стране и принимали участие преподаватели МФТИ.

Приводится доклад по выполненым работам технологии и теории программирования для информационных и интеллектуальных систем, ставшим важным направлением на правительственному уровне.

Научный доклад
доктора физ.-мат наук, почетного профессора МФТИ
Главного научного сотрудника ИСП РАН
Лаврищевой Е.М.
на тему:

Наука моделирования и программирования сложных программных и технических систем

Научные проекты РФФИ
№ 16-01-00352,
№ 19-01-00206
2016-2021

15 февраля 2022

Р а з д е л 3

Перспективная разработка общего сборщика информационных и интеллектуальных ресурсов в Интернет для пользования до 2035

В качестве общего вывода по рассмотрению операций сборки в п.1-5 можно сделать вывод о том, что приведенные операции сборки (link, make, config, assembling, building, weaver, integration) ресурсов в системных средах Интернет являются базовыми средствами и можно сделать общий «сборщик» готовых ресурсов для прикладных, сервисных, информационных и технических систем в сетевой Глобальной сети Интернет.

Рассмотренные операции сборки (link, make, config, assembling, weaver) ресурсов в приведенных заданных системных средах Интернет (Corba, VS.Net, BSN, GNU, Grid, Etics) имеют схожие операционные способы сборки разных вариантов ресурсов: модулей в разных ЯП, исходных и выходных текстов компиляторных программ, сервисных и системных ресурсов Веб среды Интернет, технических средств компьютеров имеют похожие способы сборки, которые повторяются в разных общесистемных средах. Как бы не назывались способы сборки, семантика сборки остается одинаковой. Способ сборки зависит от данных, которые используются при обмене данными между связываемыми разнородными ресурсами.

Рассмотрены средства генерации общих типов данных стандарта ISO/IEC 11404 GDT-2012 требуют создания набора новых примитивных функций для нестандартных типов данных (портфелей, контейнеров, шаблонов, указателей, неструктурных данных, больших данных и др.) и использоваться в названных способах сборки разнородных ресурсов Интернет.

Для создания общего «сборщика» готовых ресурсов в прикладные и технические (макроконвейерные) системы в Интернет, необходимо на базе конфигурационной сборки реализовать следующие задачи:

1. Определить формальный вариант описания операции сборки ресурсов

2. Создать библиотеку примитивных функций для всех систем Интернет и дорабатывать примитивные функции для новые ТД (контейнеров, шаблонов, указателей и др.) согласно стандарта GDT и GPD -2012.

3. Создать анализатор операций сборщика и взаимодействия компиляторных, системных, прикладных, технических, операционных систем с учетом всех типов ресурсов.

4. Сделать анализатор соответствия типов ресурсов и взаимодействия с другим ресурсом через интерфейс и осуществлять обращение к соответствующим программам преобразования обмениваемых данных для генерации соответствующего преобразования по теории преобразования типов данных стандарта GDT и GPD.

5. Реализовать конфигурационную сборку, выполняя все процессы, определенные в стандарте IEEE 828 Configuration:2012.

6. Выдавать сведения о результатах сборки и завершения работы сборщика.

7. Обеспечить размещение ресурсов в библиотеках и хранилищах Интернет из разных предметных областей знаний (медицины, биологии, генетики, математики и др.).

Техническим результатом способа сборки общего сборщика является:

- простота поиска ресурсов в хранилищах Интернет и снижение затрат на разработку за счет готовых правильных многоразовых ресурсов и настройку их на конкретные условия применения;
- повышение качества и производительности создаваемых из ресурсов Веб-приложений и систем за счет системных операций замены отдельных более правильных ресурсов и изменения конфигурации (архитектуры) варианта конфигурационного файла с получением качественных решений функциональных задач приложений в разных предметных областях знаний.

Создание методом сборки экспериментального варианта ОС Linux 2018-2022*

Проектирование любой системы - это процесс определения архитектуры, компонентов, интерфейсов, других характеристик системы и конечного состава программного продукта.

Базовая концепция проектирования ПО - это методология проектирования архитектуры с помощью разных методов (объектного, компонентного и др.), процессы ЖЦ (стандарт ISO/IEC 12207) и техники - декомпозиция, абстракция, инкапсуляция и др.

Ключевыми вопросами проектирования ПС является: декомпозиция программ на функциональные компоненты для независимого и параллельного их выполнения, принципы распределения компонентов в среде выполнения и их взаимодействия между собой, механизмы обеспечения качества и живучести системы и др.

При разработке операционной системы на базе Linux основным параметром, определяющим необходимый функционал, является область применения этой ОС.

Функциональные элементы ОС Linux отвечают за определенную функциональную деятельность, а именно:

- видеопамять и работа с ней;
- HAL (Hardware Abstraction layer) - поддержка нескольких аппаратных архитектур, включая процессы, семафоры и.др.:
- управление памятью;
- управление исполнением управление устройствами;
- виртуальные файловые системы;
- API (Application Programming Interface) IPC (Interprocess Communication)

и т.д.

В маленьких встроенных системах (сайтах), не будут использоваться функциональные модули, отвечающие за видеопамять, API или управление исполнением.

Процессы определения варианта ядра ОС Linux

Для конфигурации варианта ОС необходимо выполнить 6 процессов **следующего вида**:

1) **Конфигурирование.** Во время сборки Linux проверяет файл конфигурации Kconfig на наличие рекурсивных зависимостей и несуществующих переменных в коде

2) **Поиск «мертвого кода»** т.е. такого кода, в котором контроль не передается ни при каких обстоятельствах.

3) **Препроцессирование – это** предварительная обработка элементов функции из ядра ОС, которое проводится перед компиляцией для получения кода версии программы или системы. Если находятся ошибки, то выдается соответствующая информация для мсправления.

4) **Компиляция** состоит в выдаче кода или случайных ошибок при обнаружении необъявленной переменной / функции, отсутствие пунктуации в коде и т. д.

5) **Связывание** отдельных элементов ядра выполняет оператор Link. Он находит ошибки задания связи компонентов в интерфейсе или ошибки вызова компонентов из внешних библиотек.

6) **Обработка** ошибочных ситуаций.

7) Определение конфигурации проводится с целью разделения конфигурационного пространства на классы эквивалентности и использования критерия охвата тестирования MC / DC с помощью двух основных понятий: *решение* и *условие*. Решение - это формула, состоящая из условий и передач управления блоку с решением. Условие - это логическая часть решения, которое соединяется с другими условиями. Каждое условие может влиять на итоговое значение принимаемого решения, фактически изменяет значение независимо от других условий.

8)**Тестирование** собранного варианта ОС для обработки ошибок с помощью LDV и CPAchecker. Инструмент LDV ставит метки кода в соответствии с заданными правилами, CPAchecker проверяет доступность меток и правильность выполнимости компонентов.

Сборка экспериментального ядра OS Linux

Создание варианта ядра OS Linux д, химии ля класса прикладных систем (медицины, биологии и др.) основывается на анализе базовых функций ядра ОС и выборе из множества компонентов ОС наиболее подходящих для оперативного управления прикладными системами. Исходя из публикаций установлено, что OS Linux содержит более 10 000 переменных и большое множество функциональных системных компонентов, обеспечивающих обработку разного рода заданий по функционированию любых прикладных систем. На ее основе выбраны необходимые компоненты ОС и создана модель MF с базовыми характеристическими компонентами ОС и модели системы *Msys,* включая множество функциональных M_f, интерфейсных M_{io} и M_d и работы с данными базового ядра ОС. Эти множества компонентов были протестированы на правильность их выполнения, на наборах тестов и операций установления связей с соответствующими компонентами других множеств. После тестирования используются операции *config* (M_{fi}, MI_{oi}, M_{di}) для получения конфигурационного файла экспериментального варианта ядра OS Linux. Процесс формирования варианта ОС проводился с помощью выбранных операционных функциональных компонентов [4-8] и операций:

- общей и сетевой настройки,
- загруженных модулей и блоков,
- драйверов устройств и файловых систем,
- операций безопасности и криптографии,
- библиотечных процедур, компонентов и др.

При создании конфигурации ядра варианта ядра ОС используются эти параметры в зависимости от области предназначения класса прикладных систем. Например, ядро может включать в себя множество опций безопасности исходя из стека SELinux Национального

агентства по безопасности NSA, ориентированных на безопасность функциональных компонентов ОС.

Перед проведением конфигурационной сборки варианта системы ОС проверяется файл /etc /udev /rules.d/70-persistent-net.rules и определяются имена сетевых устройств, а также схема именования правилами Udev. Выходной файл имеет блок комментариев, за которым следуют две строки для каждого сетевого адаптера. Первая строка сетевой карты включает описание и комментарии, показывающие идентификаторы оборудования и устройств согласно карты PCI и драйверов.

При задании имени интерфейса идентификатор аппаратного обеспечения не используются. Вторая строка - это правило Udev, которое соответствует сетевой карте с фактически присвоенным именем.

Некоторое программы ПО ОС могут устанавливаться с помощью символических ссылок /dev /cdrom и /dev/dvd для устройств CD-ROM или DVD-ROM. Кроме того, могут помещаться символические ссылки в /etc/fstab. Udev в зависимости от возможностей каждого устройства.

Сценарий может работать в режиме «by-path» по умолчанию для устройств USB и FireWire, создаваемые правила которых зависят от физического пути к устройству CD или DVD. Сценарий может работать в режиме «по-id» (по умолчанию для устройств IDE и SCSI) и зависит от правил идентификационных строк, хранящихся на устройстве CD или DVD. Физический путь к устройству (порты и / или слоты) изменяются, например, при планировании перемещения диска в другой порт IDE или разъем USB режима «by-id».

Для каждого устройства находится соответствующий каталог в разделе / sys / class или / sys / block и для видеоустройств в разделе / sys / class / video4linux / videoX.

Интерфейсы сетевых сценариев задаются в файле / etc / sysconfig /. При этом файл ifconfig содержит настраиваемый интерфейс ifconfig.xyz в сетевой карте с именем eth0. Внутри этого файла содержатся атрибуты интерфейса IP-адрес, маски подсети и т. д. Приводится спецификация выходного файла в ЯП.

К этой работе привлевались студенты МФТИ, с их участием сделан экспериментальный вариант ядра OS Linux.

Фабрика программ к 90-летию академика Глушкова

Концепция сборочного конвейера Глушкова реализована студентами 4 курса факультета кибернетики КНУ имени Тараса Шевченко в виде фабрики программ (веб-сайт http://programsfactjry.univ.kiev.ua) под руководством автора. Фабрика – составная часть ИТК ИПС [9] (http://sestudy.edu-ua.net). Она оборудована линиями, созданными студентами: программирование в языках. C # VS.Net, Java, DSL; построение артефактов для информационных и программных систем с их сохранением в репозитории; сборка программных компонентов в сложные структуры ПС; трансформация передаваемых типов данных; метрический анализ и оценка качества ПС; обучение теоретическим и прикладным аспектам SE по е-учебнику. Основу этой фабрики составляют готовые компоненты повторного использования. Фабрика практически работает в Интернете с декабря 2011г. К ней обратилось более 10 000 пользователей из разных стран. Фабрика пополниться новыми линиями: генерация прикладных систем в языке DSL (Domain Specific Language); трансформация общих типов данных GDT стандарта ISO/IEC GDT к фундаментальным FDT; построение распределенных ПС из сервисов и др. Идея сборочного конвейера на десятилетия определила ход развития индустрии ПП и фактически студентами сделан подарок к 90-летию Глушкова.

Литература

1. *Капитонова Ю.В., Летичевский А.А.* Парадигмы и идеи академика В.М. Глушкова.– Киев, Наук. Думка.– 2003.–355 С.

2. *Глушков В.М.* Кибернетика, ВТ, информатика (АСУ).–Избр. труды в 3-х томах. –К.: Наук. думка, 1990, 262 С, 267 С., 281 С.
3. *Глушков В.М.* Фундаментальные основы и технология программирования //Программирование, 1980.–№2.–с. 3–13.
4. *Глушков В.М.* Основы безбумажной информатики.– М.: Наука, 1982. – 552 С.
5. *Глушков В.М., Лаврищева Е.М., Стогний А.А*. и др. Система автоматизации производства программ (АПРОП). – Киев: Ин – т кибернетики АН УССР, 1976. – 134 С.
6. *Глушков В.М., Капитонова Ю.В., Летичевский А.А.* О применении метода формализованных технических заданий к проектированию программ обработки структур данных.– Программирование, 1978, №6, с.31–40.
7. *Вельбицкий И.В., Ходаковский В.Н., Шолмов Л.И.* Технологический комплекс автоматизации программ на машинах ЕС ЭВМ и БЭСМ–6. М.: Финансы и статистика.–1980.–253 с.
8. *Лаврищева Е.М., Грищенко В.Н.* Связь разноязыковых модулей в ОС ЕС.–М.: Финансы и статистика, 1982. – 127 С.
9. *Лаврищева Е.М., Зинькович В.М., Колесник А.Л.* и др. Инструментально-технологический комплекс разработки и обучения приемам производства программных систем, (укр.).- Государственная служба интеллектуальной собственности Украины.– Свидетельство о регистрации авторского права.– № 45292, от 27.08.2012.-103 с.

Про сборку Катерины Лаврищевой и Нади Мищенко

Глушков В.М. в своих выступлениях всегда высказывал много новых идей, которые подхватывали заинтересованные специалисты. ЛЕМ всегда ходила с тетрадью и записывала интересные для его мысли, особенно, если они касаются вопросов программирования или направлений, связанных с развитием новых машин «Мир», систем АСУ, АСУ ТП и др.

Одно из таких выступлений было сделано на семинаре 5 марта 1974 года в его отделе под номером 100, посвященном перспективам программирования в нашей стране.

На семинаре присутствовали Ющенко Е.Л., Сергиенко И.В., Летичевский А.А., Капитонова Ю.В., Молчанов И.Н., Лаврищева Е.М., Никитин А.И., Бабенко Л.П. и др.

Глушков сказал: «Пройдет 20-30 лет и сложные программы будут выпускаться, как на сборочном конвейере Форда из готовых «деталей». Появятся фабрики программ, работающие про принципу сборки продуктов из готовых программ, как в автомобильной промышленности».

Технология создания программных продуктов

В 1975 г. В.М. Глушков предложил перспективный способ для постепенного перехода от «ремесленного» производства к промышленному выпуску компьютеров, программ и аппаратно-программных систем. Индустрия компьютерных программ по его замыслу должна была базироваться на технологических линиях конвейерного изготовления товаров, продуктов. Технологии создания компьютеров, информационных и программных систем Глушков В.М. считал движущей силой прогресса фундаментальных кибернетических и компьютерных наук. ….

В работе [7] 1980г. он сформулировал три перспективных направления технологии программирования:

- модульная система автоматизации производства программ (АПРОП) из стандартизированных программных заготовок в сложные системы [8];

- метод формализованных технических заданий для проектирования сложных программных комплексов с использованием нескольких алгоритмических языков для описания отдельных блоков систем на уровнях последовательной детализации компьютерного проекта МАЯК [1-3];

- Р-технология программирования для автоматизации систем средствами графического Р-языка для представления структур программ и данных в АСУ [10].

В результате поисковых и прикладных исследований на направлениях автоматизации были разработаны методы, технологии, инструментальные средства. К их числу относятся: Р-технология (Вельбицкий И.В.); сборочная технология из разнородных модулей и интерфейсов АПРОП (Лаврищева Е.М.); система «Проект» (Капитонова Ю.В., Летичевский .А.), технология систем и пакетов прикладных программ (Молчанов И.Н.); пакеты прикладных программ математического, экономического, статистического типов (Сергиенко И.В., Редько В.Н., Стукало А.С.). Этими работами был внесен весомый вклад в индустрию сборочного создания программных продуктов на ЕС ЭВМ.

Сформировалась технология программирования и новый вид программирования – сборочное программирование для объединения разнородных модулей средствами системы автоматизации программ АПРОП под руководством В.М.Глушкова.

Технология сборочного программирования формировалась в отделе Глушкова по тематике системы Проект в плане создания семейства трансляторов с широко используемых языков программирования.. Основные положения этого вида сборки базировались на идеи выделения общих средств в языках программирования ОС ЕС, системной реализации компонентов языковых процессоров и сборкой из них трансляторов в системе ТЕРЕМ (Мищенко Н.М.).. В этой системе разработаны общие компоненты для языковых процессов в классе языков ОС ЕС и представлены в мультипроцессорном комплексе МАЯК в работе:

Н.М.Мищенко. «О сборочном программировании языковых процессоров». - Сб. Интеллектуализация программного обеспечения информационно-вычислительных систем.– К: Ин-т кибернетики им. В.М. Глушкова АН УССР, 1990. – С. 45-52.

Система АПРОП разрабатывалась по договору с Институтом МНИИПА (Москва) в составе технологии создания программ для бортовых систем «ПРОТВА», реализованной под руководством В.В. Липаева [12-15]. Главное нововведение этой системы – интерфейс (межмодульный, межъязыковый и технологический) [8] и библиотека интерфейсных из :64 функций преобразования нерелевантных типов данных, описываемых в модулях на разных языках и для разных платформ. При реализации задачи сборки больщих программ из готовых разноязыковых модулей впервые нами было сформулировано понятие *интерфейса* и языка его описания (1976г.) [12-14]. Идеи, заложенные в интерфейс, как способа связи разноязычных модулей и реализованные в системе АПРОП опередили появление зарубежных языков интерфейсов MIL API, IDL, SIDL и других. Интерфейс стал необходимым средством интеграции новых прикладных систем из готовых компонентов повторного использования (КПИ) и обеспечения взаимодействия программ и систем в современных глобальных и сетевых средах.

Развитие сборочного программирования поддерживали академик АН СССР А.П. Ершов и проф. В.В. Липаев в проекте «Протва» [15, 16]. Ершов А.П. считал, что «сборочное программирование - эффективно, поскольку готовые запрограммированные модули позволяют быстро решить любые задачи из определенной проблемной области для ЕС ЭВМ и мини-, микро- и макро - ЭВМ».

В дальнейшем сборка стала важным технологическим решением для индустрии создания программных продуктов, как продукции производственно-технического назначения в СССР. На многих конференциях по технологии програмирования высветливалась задача сборки специалистами, которые создавали большие программные системы для разных применений. Отшлифовывался метод сборки и в системе ТЕРЕМ при создании нового класса систем - семейства языковых процессоров для широко используемых языков программирования ОС ЕС. Метод сборки и подход к реализации метода сборки разноязыковых программ были опубликованы в зарубежной прессе Springer Modular design of large programs E. M. Lavrishcheva 1980, Volume 16, Number 2, Pages 244-249,

защищено Е.М. Лаврищевой в докторской диссертации «Методы, средства и инструменты сборочного программирования» (1989), которую оппонировали известные в СССР специалисты – Э.Х.Тыугу, Э.З.Любимский и И.В.Вельбицкий. Были опубликованы монографий «Сборочное программирование» (Лаврищева Е.М., Грищенко В.Н., 1991) и «Технология сборочного программирования» (В.В. Липаев, Б.А.Позин, А.А.Штрик, 1992) Позднее были такие публикации статьях Лаврищевой Е.М. и др. сотрудников ИК .

Methods and Tools of Component Programming V. N. Grishchenko and Ye. M. Lavrishcheva 2003, Volume 39, Number 1, Pages 33-45.

Compositional programming: theory and practice K. M. Lavrischeva 2009, Volume 45, Number 6, Pages 845-853.

Theory and practice of software factories K. M. Lavrischeva 2011, Volume 47, Number 6, Pages 961-972

Formal fundamentals of component interoperability in programming K. M. Lavrischeva 2010, Volume 46, Number 4, Pages 639-652.

Reuse problems in software engineering L. P. Babenko 1999, Volume 35, Number 2, Pages 314-323.

Information Support of Reuse in UML-Based Software Engineering L. P. Babenko 2003, Volume 39, Number 1, Pages 65-70

Мищенко Н.М. "*Об одном комплексе программных модулей для сборки трансляторов.*
Аннотация . *Рассматриваемый комплекс предназначен для реализации однопроходных трансляторов на ЕС ЭВМ для класса языков, который включает языки программирования и подмножества естественных языков, используемых для общения в*

человеко-машинных системах и описываемых нелеворекурсивными контекстно-свободными грамматиками. В программных модулях комплекса реализованы хорошо изученные и наиболее часто встречающиеся в трансляторах функции перевода и структуры данных. Предполагается участие пользователя в создании непредусмотренных в комплексе модулей или "своих" версий уже имеющихся модулей по прилагаемой к комплексу методике.

Мищенко Н.М. *О сборочном программировании языковых процессоров* / Сб. Интеллектуализация программного обеспечения информационно-вычислительных систем. Сб.науч. тр. – Киев : Ин-т кибернетики им. В.М. Глушкова АН УССР, 1990. – С. 45-52.
Аннотация. *В работе рассматривается способ реализации языковых процессоров путем их сборки из готовых программных модулей, которые вместе со средствами и методикой их адаптации к конкретному реализуемому языку образуют систему сборочного программирования языковых процессоров.*

Підведено підсумки опису Теремківського періоду 1970-80-х років.
Головним моїм заняттям протягом цього періоду була побудова інструментальної системи програмування під назвою розширна система програмування (РСП) ТЕРЕМ та її застосування для розробки трансляторів для Макроконвейєра з мов, граматика яких описувалася мовою БНФ.

РСП ТЕРЕМ – це універсальна синтаксична підсистема (Аналізатор вхідних мов і Конструктор синтаксичних таблиць) та універсальні програми семантичної підсистеми, відкритої для розширення. У транслятор, який розроблявся для мови з допустимого класу мов, повністю включався Аналізатор, а по БНФ-опису граматики мови програма Конструктор будувала синтаксичні таблиці, які з Аналізатором складали синтаксичну підсистему транслятора.
У складі семантичної підсистеми виділялися засоби систем побудови трансляторів (СПТ). Їх наявність у складі новоствореної системи програмування дозволяла програмістові змінювати семантику вхідної мови в процесі використання транслятора. Відсутність засобів СПТ в трансляторі означала, що вхідна мова фіксована і не потребує розвитку під час її використання. Можливий третій варіант: включення засобів СПТ у транслятор для розвитку вхідної мови до певного рівня, після чого засоби СПТ можут бути вилучені з транслятора.
Оригінальним був зв'язок синтаксичної та семантичної підсистем: до елементів опису синтаксису мовою БНФ дописувалися імена відповідних семантичних дій з ознаками, які регулювали час ініціації дій над фразою, розпізнаною під час синтаксичного аналізу.
Здатність вхідної мови до змін була необхідна для реалізації нових мов, що еволюціонують в процесі їх використання, як це було у випадку з мовами програмування для Макроконвейєра.
Частина виступів на семінарах та конференціях стосувалася загального опису використання РСП ТЕРЕМ, інша частина висвітлювала окремі властивості трансляторів, побудованих за допомогою РСП ТЕРЕМ, завдяки яким вдавалося реалізувати:
- дослідницький характер реалізації мови, якщо в її транслятор включалися засоби СПТ;
- адаптовність системи до конкретних умов обчислювальної системи;
- практичність системи, оскільки 3/5 нового транслятора – це засоби РСП ТЕРЕМ.

Протягом 1990 року пощастило перекласти РСП ТЕРЕМ, запрограмовану мовою ПЛ/1, на мову Сі для роботи на персональних комп'ютерах (ПК). Переклад мовою Сі виконувався буквально оператор за оператором, опис за описом. Яке ж було моє здивування, коли на першому ж сеансі роботи на ПК РСП ТЕРЕМ запрацювала!

Таким образом, сборочное программирование сложных систем из готовых программных ресурсов было де-факто сформировано как новый вид программирования и основа фабрик программ конвейерного типа по ТЛ в соответствии с концепцией академика Глушкова.

Новый вид программирования – сборочный, ориентирован на объединение разноязычных модулей, которые специфицировались на разных языках ОС ЕС (PL/1, Fortran, Algol-60, Cobol, Modula / 2, Assembler). Средствами их объединения в большой ОС ЕС были: интерфейс, модули

повторного использования в МП, жизненный цикл технологии сборки, и система сборки (сборочный конвейер) готовых модулей и программ в более сложные программные структуры. Система АПРОП разрабатывалась при финансовой поддержке министерства радиопромышленности СССР более 10 лет и ставшей частью технологии «Протва» В.В.Липаева для бортовых машин. Систему АПРОП использовалась для сборки программ более, чем 52 организаций СССР.

Главное новшество в сборочном программировании – интерфейс (межмодульных, межъязыковой и технологический) [19] и библиотека интерфейсных функций преобразования типов данных с одного ЯП к другому средствами системы АПРОП. Первое определение понятия интерфейса и языка его описания сформулированы нами в проекте этой системы в 1976г. под руководством Глушкова [2]. Идея интерфейса для связи модулей намного опередила в этом направлении зарубежные разработки. Язык MIL (Module Interface Language) появился там в 1983г. В настоящее время интерфейс является актуальним и определяется с помощью программного интерфейса API (Application programs Interface), интерфейса IDL (Interface Definition Language) и др.

Интерфейсу была посвящена международная конференция «Интерфейс СЭВ» (1987) [24]. На ней были представлены концепция интерфейсов и система АПРОП. Организатор этой конференции ГКНТ СССР наградил коллектив разработчиков из трех специалистов (Коваль Г.И., Коротун Т.М., Лаврищева К.М.) почетной грамотой.

Важную роль в сборочном программированию сыграл академик А. П. Ершов. В докладе «Научные основы доказательного программирования» на президиуме АН СССР (1984) отметил следующее.

«Сборочное программирование решает задачи многократного и быстрого применения в процессе создания программы с заранее изготовленных «деталей. Оно эффективно, когда комбинирование сравнительно небольшого числа заранее запрограммированных модулей, позволяет быстро решить любую задачу из некоторого класса часто возникающих проблем. Ориентация на класс задач – особенность сборочного программирования, что объясняет его актуальность, поскольку широкое распространение мини – и микро-ЭВМ позволяет применять каждую отдельную машину для решения определенных специальных задач»

В другом докладе А.П. Ершова «Отношение методологии и технологии программирования» [25] на Всесоюзной конференции по ТП (1986р.) сформулированы перспективы промышленной сборочной технологии программирования до2005г. включая нормативы производительности и надежности продукта, этапность разработки ПП и межмодульных интерфейс поддержки сборочного программирования».

ТЛ как элемент сборочного конвейера, определен нами при участии в разработке программного обеспечения АИС «Юпитер-470» для четырех объектов флота. Для каждого объекта были разработаны десятки типовых программ обработки данных [24] благодаря созданным шести ТЛ с изготовления различных видов программ, необходимых при решении практических задач на объектах АИС. Эти ТЛ – первый вариант автоматизированного сборочного конвейера Глушкова. С их помощью были созданы около 500 программ обработки данных для объектов АИС.

Концепция сборочного конвейера Глушкова реализована с участием студентов факультета кибернетики КНУ и МФТИ в виде экспериментальной фабрики программ (веб-сайт http://programsfactjry.univ.kiev.ua).

В рамках проекта разработана теория, методология и практика изготовления ПП в соответствии с методикой представленной в журнале «Кибернетика и системный анализ 2010» и в Springer Theory and practice of software factories K. M. Lavrischeva, 2011, Volume 47, Number 6, Pages 961-972, а также изложенной в учебниках по программной инженерии, участвующих на

конкурсах фирмы Майкрософт в Москве-2006 (русский вариант опубликован в www.intuit.ru) и на конкурсу в Киеве-2007, получен диплом Лаврищевой Е.М. по номинации «Учитель-новатор» за учебник SE укр. [20].

Фабрика программ демонстрировалась на международном научном конкурсе Агенства по информатизации при правительстве «Информационное общество в Украине» 2012г.

Многие идеи фабрики программ вошли в проект инструментального комплекса ИТК ИПС http://sestudy.edu-ua.net. Комплекс оборудован фабричными 4 линиями, созданными студентами и 7 линиями, изготовленные аспирантами. К ИТК (2012-2013) обратилось, как показывает google статистика более 15000 пользователей из разных стран СНГ, США, Бразилии, Португалии, Китай и др. .

Сайт В.П.Глушкова

2.1. Способы сборка систем из модульных элементов через сборшик ресурсов Интернет

В 1975году академик В.М.Глушков высказал идею о том, что «программы будут собираться конвейерным способом, как автомобили из готовых деталей на фабрике Форда». При исследовании этой идеи сформировалось понятие интерфейса связывания программных элементов (по типу болтов и гаек деталей автомобилей) методом сборки, который объединял разнородные модули через интерфейс. Межмодульный интерфейс реализован в 1976–1982 г. в системе АПРОП [11,12]. Межъязыковый и технологический интерфейс обсуждался на

конференции «Интерфейс СЭВ-1987 и стал базовым понятием в технологии программирования. Автор получил грамоту Евроконференции Интерфейс СЭВ за определение понятия интерфейса рис.2).

Метод сборки и библиотека интерфейса (64 примитивов преобразования неэквивалентных данных) реализованы в ОС ЕС или IBM 360, переданы в 52 организации СССР и внедрены в комплексы РУЗА, ПРОМЕТЕЙ, ЯУЗА (В.В. Липаев, МНИИПА, Минрадиопрома СССР) для специализированных ЭВМ ВПК.

Интерфейс модулей в языке MIL (Module Interface Language), а операция сборки модулей задавалась оператором link (M1, M2) модулей. Метод сборки с интерфейсными примитивами преобразования разнородных данных ЯП опубликован в книге [12] Эта система была внедрена в 52 организации страны. Сформировалось сборочное программирование [12-15].

Рис.1. Интерфейс в программировании

Рассмотрим существующие виды сборки 1-5 ресурсов в ОС Интернет с использованием стандарта конфигурационной сборки IEEE 824 –Configuration: 1996.

1). Способ сборки make реализован в системах BSD, GNU, Java, как оператор объединения функций в ЯП из библиотек модулей транслятора Microsoft, UNIX и др., интерфейс которых задавался в API (Applic ation Programming Interface) на уровне исполняемых файлов в машинном коде и в ABI (Application Binary Interface) для объединения скомпилированных модулей в промежуточном MSIL байт-коде (C++, Fortran, Go, Perl, PHP, Java) в среде .NET. Процесс сборки make ресурсов в API и ABI ЯП выполняет (рис.2):

- обработку API и ABI интерфейсов из библиотеки .NET;
– генерацию сборочных скриптов GNU Build system, CMake;
– MSBuild (.NET), Apache Ant (Java), Apache Maven (Java, C#, Scala), NAnt (.NET), Scons(C, C++, Java, Fortran,Tex).

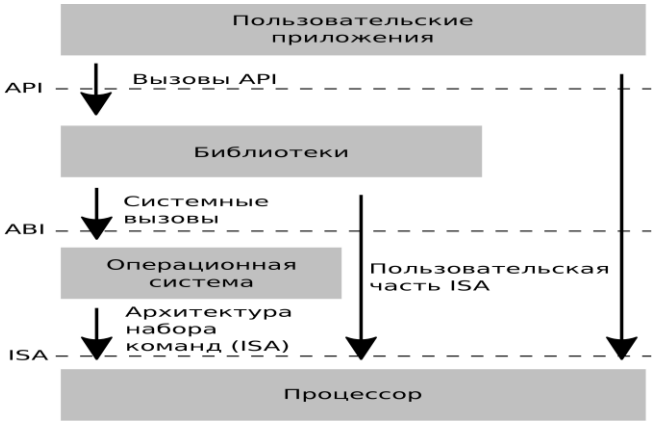


Рис. 2. Схема обработки make

Процесс сборки ресурсов в ЯП с интерфейсом в API и ABI через make выполняет генератор сборочных скриптов GNU Build system в средах MSBuild (.NET), Apache Ant (Java), Apache Maven трансляторов с Java, C#, Scala; Scons(C, C++, Java, Fortran,Tex) и др. ABI Interface содержит:

– набор инструкций процессора к регистровому файлу, стеку и памяти;

– размер и расположение базовых ТД, с которыми работает процессор компьютера;

– бинарные форматы файлов, библиотек и исполняемых файлов.

Операции связи программ в API и ABI для C++ и Fortran (рис.3):

- add executable (exec_name source1 source2 …);
- создать исполняемый файл из файлов исходного кода source1, source2, и т.д.

main.cpp

```
#include <iostream>

extern "C" void fort(void);
extern "C" void cube(float* x, float* y);

int main() {
  fort();
  std::cout << "C++\n";
  float x = 2.5, y = 0;
  cube(&x, &y);
  std::cout << x << ' ' << y << '\n';
}
```

fort.f90

```
subroutine fort() bind(C)
  implicit none
  write(*,*) "Fortran"
  return
end

subroutine cube(x,y) bind(C)
  implicit none
  real*4 x, y
  y = x * x * x
  return
end
```

Запуск

```
gfortran -c fort.f90 -o fort.o
g++ main.cpp fort.o –lgfortran
./a.out
```

Вывод

```
Fortran
C++
2.5 15.625
```

Рис.3. Пример сборки модулей в C++ и Fortran

2). Способ сборки через интерфейс Stub-клиент и Skeleton-сервера в IDL (Interface Definition Language) задают описание интерфейсов программ в языках (Java, Cobol, PL/1, Ada-95, C, C++ и др.). Их связь в IDL в разных ЯП объектной модели (ОМ) задает брокер CORBA. В рамках стандарта WWW Web появился язык WSDL (2004) для связывания разнородных программных элементов в ЯП нового поколения (Си, C++, Basic, Java, Cobol, ADA-95, Python и др.) с использованием библиотеки 64 функций в среде Интернет (рис.4)

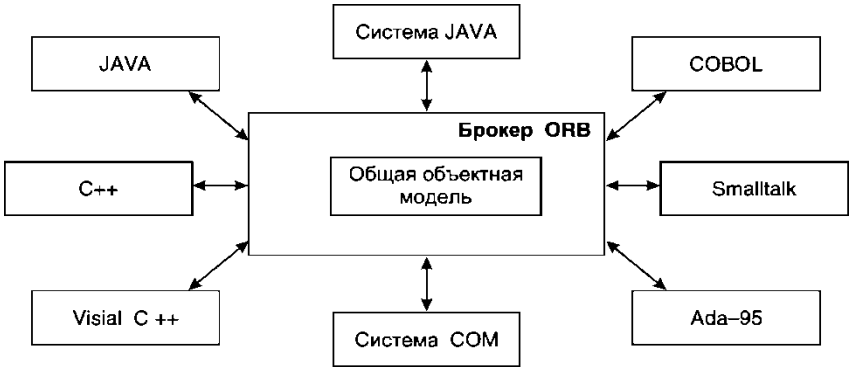
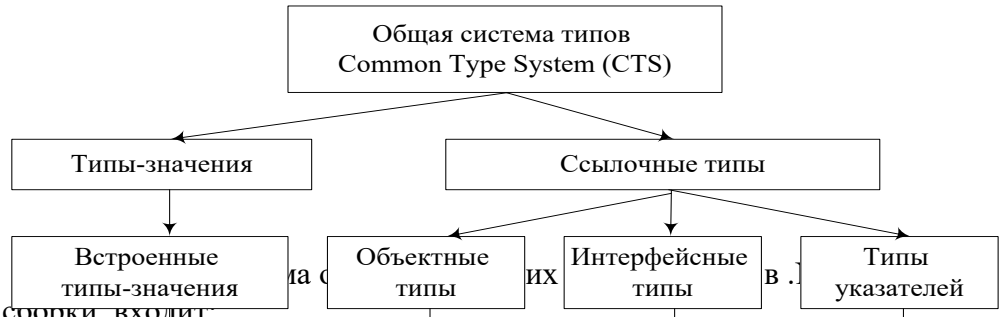


Рис. 4. Функциональная схема брокера ORB

3). Способ сборки в среде .Net основана на обработке передаваемых данных между модулями с помощью системы CTS (Common Type System, рис.5).



В задачи сборки входит:

- отобр[illegible]пьютера;
- описа[illegible]ка

целочисленных данных, с плавающей запятой, строк, битов;

- спецификации классов, интерфейсов, встроенных типов данных, перечислений и др., заданных в CLS (Common Language Specification);
- перевода данных в ЯП к промежуточному MSIL, который преобразуется в код CPU для выполнения на разных архитектурах компьютеров сети.

При сборки объектов на основе интерфейса в языке IDL описание начинается с ключевого слова interface, за которым следует: имя интерфейса, описание типов параметров и операций (op_dcl) вызова объектов:

```
interface A { ... }
interface B { ... }
interface C: B, A { ... }.
```

Параметры операций (op_dcl) в задании интерфейсов это:

- тип данных (type_dcl);
- константа (const_dcl);
- название исключительной ситуация (except_dcl), которая может возникнуть в процессе выполнения метода объекта;
- атрибуты параметров (attr_dcl).

Описание типов данных (ТД) начинается ключевым словом typedef, за которым следует базовый или конструируемый тип и его идентификатор. В качестве константы может быть некоторое значение типа данного или выражение, составленное из констант. ТД и константы описываются как фундаментальные типы данных: integer, boolean, string, float, char и др.

Описание операций op_dcl передачи данных включает в себя:

- наименование операции интерфейса;

- список параметров (от нуля и более);
- типы аргументов и результатов, иначе – void;
- управляющий параметр или описание исключительной ситуации и др.

Атрибуты передаваемых параметров начинаются служебными словами: in – при отсылке параметра от клиента к серверу; out – при отправке параметров-результатов от сервера к клиенту; inout – при передаче параметров в оба направления (от клиента к серверу и обратно).
Описание интерфейса для одного объекта может наследоваться другим объектом и тогда это описание становится базовым. Пример такого дан ниже:

```
const long l=2
interface A {
void f (in float s [l]); }
interface B {
const long l=3 )
interface C: B, A { }.
```

Интерфейс С использует интерфейс В и А. Это означает, что интерфейс С наследует описание типов данных А и В, которые по отношению к С являются внешними. Но при этом синтаксис и семантика остаются неизменными. Согласно приведенного примера - операция функции f в интерфейсе С наследуется из А.

4). Способ сборки в GRID (www.gloubos.org) обеспечивает взаимодействие ресурсов через вызовы RPC/RMI в программах на ЯП с помощью операций assembling, config, make и устанавливают связь ресурсов между собой при работе с Cloud Computing и Big Data. Система GRID (рис. 8) обеспечивает обработку и управление программными, сервисными и др. Web ресурсами глобальной сети.
Сборка разнородных сетевых и системных ресурсов в Веб системы, приложения и пакеты, которые работают с данными разного объема, проводится в системе ETICS GRID (рис.6).

Ресурсы описываются в разных современных ЯП. Базовые сущности систем, пакет и приложений, а также связи описываются в CIM (Common Information Model).
В системе ETICS используется стандартизованное описание ТД для главных объектов: Проект, Подсистема и Компонент. Проект. Подсистема может содержать только Компоненты. В них используется модель данных CIM, задающая связь между разными объектами и описывать объекты и связи между ними, а также передавать результаты их выполнения по запросам.
Сохранение и ведение данных базируется на модели реляционного типа в MySQL.
Описание модели данных основано на следующих базовых положениях:

1) каждый компонент содержит описание сведений (имя, лицензия, URL репозитория и т. п.) и глобального уникального идентификатора – ID (GUID);

2) объект конфигурации содержит информацию о версиях, связи с репозиторием, GUID, платформе фреймворка и связь с конфигурацией системы;

3) каждый компонент проходит проверку (checkout) скомпилированного элемента, тестовых команд и GUIDs, а также связь с конфигурацией;

4) при определении конфигурации и платформы в каждом объекте появляется GUID, его свойства, среда выполнения и зависимости, которые могут объявляться статически или динамически. Статическая зависимость – это взаимоотношение между двумя конфигурациями, динамическая зависимость – взаимоотношение между конфигурацией и модулем.
ETICS по функциям соответствует концепции современной фабрики программ. Она базируется на наборах характеристик, услуг и процедур изготовления пакетов.
Пакеты могут объединяться плагинами с услугами для потребителей и поставщиков, обеспечивать управление заданиями с рабочих мест, а также давать доступ к OCAM, архитектуре CPU, компиляторам в ЯП и средствам спецификации зависимостей между разными пакетами и их

тестами при сборке программ и их развертывании. Множество функциональных плагинов обеспечивает проверку контрактов, тестов выполнения разных элементов систем, генерацию документации, ведения готовых КПИ в оперативном или статическом репозитории ETICS.

Главная задача системы ETICS состоит в преобразовании некоторых компонентов систем для альтернативной платформы гетерогенной среды компьютеров методом ссылок с 16-, 32-разрядных платформ на 64-разрядную платформу среды Grid.

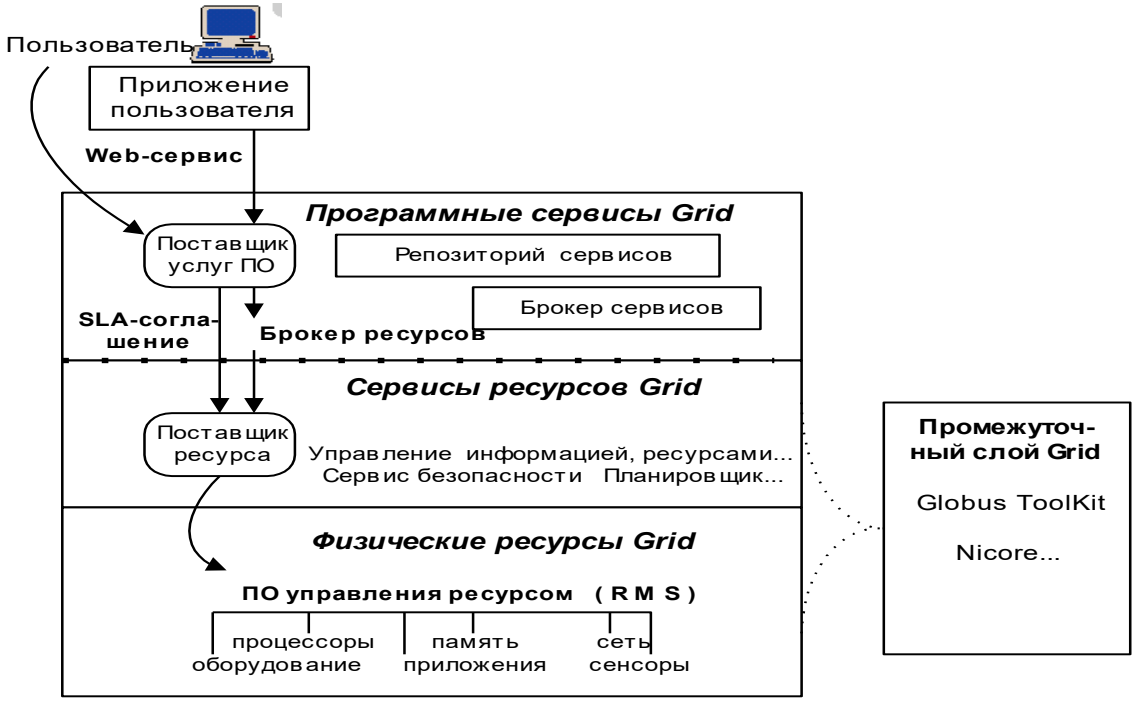


Рис.6. Общая структура функционирования GRID

Вопросами поддержки интерфейса ПО занимается UNICORE (UNiform Interface to COmputing REsources) (www.unicore.org). Обеспечение безопасности и защиты данных ресурсов и систем осуществляет инструмент WSRF (Web Service Recourse Framework).

Вопросами поддержки интерфейса ПО занимается UNICORE (UNiform Interface to COmputing REsources) (www.unicore.org). Обеспечение безопасности и защиты данных ресурсов и систем осуществляет инструмент WSRF (Web Service Recourse Framework).

5) Способ сборки по типу фабрик программ в IBM WebSphere, Microsoft Biz Talk, BEA WebLogic Oracle 10g, SAP NetWeaver и ИВК «Юпитер». В них содержатся CASE-средства сборки (link) ресурсов (сервисов, компонентов и данных) через брокера обмена данными stub и skeleton системы CORBA, переедаемых данные с помощью протокола Workflow с проверкой безопасности и качества (табл. 1) [24].

Таблица 1. CASE–системы сборки ресурсов

Платформа	Фирмы	Содержание платформы
IBM WebSphere	Корпорация IBM	Сервер приложений J2EE, брокеры обмена данными, КПИ, портал, workflow/BPM, EII, SCA
Microsoft Biz Talk 2004 и компоненты .Net	Корпорация "Майкрософт"	Сервер приложений COM, брокеры обмена данными, RGB доставка, портал, workflow/BPMN

BEA WebLogic	Корпорация "BEA Systems" (с 2008г. входит в "Oracle")	Сервер приложений J2EE, брокеры обмена данными, ГОР, сервер прикладных приложений, портал, workflow/BPMN
Oracle 10g	Корпорация "Oracle" http://www.oracle.com	Сервер приложений J2EE, брокеры обмена данными, ГОР, портал, workflow/BPMN, средства EII
SAP NetWeaver	Корпорация SAP http://www1.sap.com/www.sap.ru	Сервер приложений J2EE/ABAP, брокеры обмена данными, портал, инструменты BPMN
ИВК "Юпитер	Компания ИВК (Россия) http://www.ivk.ru/	Брокеры обмена данными, КПИ, среда выполнения, сертификация, защита данных

На фабриках программ этих систем модифицируется архитектура в соответствии с требуемыми кодами сервисных ресурсов модели SOA в Visual Studio Industry Partners (VSIP). При этом используются модели Guidance Automation eXtensions (GAX) и Guidance Automation Toolkit (GAT) в Visual Studio. GAX — это среда исполнения в VSIP с использованием пакетов рекомендаций. Существует два варианта служб: веб-служба ASP.NET (ASMX) для Windows Communication Foundation (WCF) в среде .NET Framework 3.0. Версии для веб-службы WCF находятся у разработчиков фабрики ПО GotDotN. В ее состав входят процессы предприятия и пакеты документаций и рекомендаций для приложения типа Global Bank. Это аналогично модели SCA в IBM WebSphere.

6). Способы автоматизированной сборки (интеграции) программ на CASE инструментах: Make, Apache Ant, Apache Maven, Gradle и др.

Make — это кросс-платформенная система автоматизации сборки систем из исходного кода. Она генерирует файлы управления сборкою, например Makefile в системах Unix для сборки посредством операции make (см. п.4.)

Apache Ant — JAVA-утилита для автоматизации процесса сборки программного продукта. Ant — платформо-независимый аналог UNIX-утилиты Make, но с использованием языка JAVA, приспособленного для JAVA-проектов. Самая важная разница между Ant и Make состоит в том, что Ant использует язык XML для описания процесса сборки и его зависимостей, а Make имеет свой собственный формат файл Makefile. XML-файл (или build.xml) осуществляет сборку исполняемых программ.

Apache Maven — программный инструмент для управления (management) JAVA-проектами методом сборки (building) ПО аналогично Apache Ant, но с более простой build-моделью конфигурации, которая базируется на формате XML. Двигатель ядра инструмента динамически загружает плагины из репозитория и обеспечивает доступ ко многим версиям разных JAVA-проектов с открытым кодом Apache.

Gradle — система автоматической сборки, построенная на принципах Apache Ant и Apache Maven, но вместо XML-подобной формы здесь используются языки DSL и Groovy для представления конфигурации создаваемого приложения или проекта.

Создание Общего сборщика разнородных ресурсов в Интернет

В качестве общего вывода по рассмотрению операций сборки можно сделать вывод о том, что приведенные операции сборки (link, make, config, assembling, building, weaver, integration) ресурсов в системных средах (3.2 -3.9), являются базовыми средствами для создания общего «сборщика» готовых ресурсов для прикладных, сервисных, информационных и технических систем в сетевой Глобальной сети Интернет. Основу сборщика ресурсов составляет теория преобразования сложных типов данных стандарта ISO/IEC 11404 GDT к более простым, с

которыми могут вычисляться любые прикладные системы. Далее рассматриваются общие положения теории преобразования типов данных стандарта ISO/IEC 11404-2012 GDT.

Теоретический базис преобразования ТД FDT, GDT , GLD

Разнородные КПИ, сервисные компоненты работают с данными, которые включают множество значений, операции над этими значениями и взаимодействие выполнения операций над ними. Первоначально в методе сборки модулей использовалась аксиоматика FDT (Fundamental Data Types) в классе ЯП для ОС ЕС. Эта аксиоматика разработана известными специалистами Э. Дейкстрой, Н. Виртом, В. Турским, В.Н. Агафоновым и др. в 70-годы [5, 16]. Позднее в 1996 г. разработана аксиоматика общих типов данных (General Data Types — GDT) в стандарте ISO/IEC 11404-GDT, 1996, 2007.

К фундаментальным типам данных (FDT) относятся:

- простые типы (integer, real, boolean, character, bite и др.);
- сложные (массивы, таблицы, файлы, записи, множества, деревья и др.).

Эти ТД задают базовые значения данных в программах на ЯП, которые проверяется на этапе компиляции на соответствие типов. На этапе выполнения они реализуются с помощью предикатов полиморфных типов. КПИ обмениваются данными, ТД которых должны соответствовать друг другу. В случае несоответствия ТД (например, передается целое — integer, а требуется для выполнения вызываемого модуля real), проводятся эквивалентные преобразования обмениваемых данных (integer ⇔ real) с помощью примитивных функций библиотеки интерфейсов АПРОП для FDT. К проблемам преобразования ТД в разных ЯП относятся:

а) несоответствие количества (формальных и фактических) параметров или неверное их описание;

б) несогласованность типов передаваемых параметров или их значений для форматов компьютеров;

е) отсутствие прямых и обратных преобразований параметров и др.

Так как FDT не охватывал все виды данных и типов новых ЯП после 1992г., то появился новый стандарт общих типов донных GDT, которые и использовались в информационных и прикладных системах. В стандарт GDT включены FDT и новые сложные типы данных — контейнеры, указатели, множества, списки, последовательности, неструктурные данные и т.п. Данный стандарт ISO/IEC 11404 GDT — 1996 прошел многолетнюю апробацию и вышел новый вариант в 2007г. В его состав входят такие типы данных: агрегатные, генеративные, расширяемые и др., которые требуют их генерации к фундаментальным ТД для последующего применения в ресурсах Глобальной сети Интернет. Требуется разработка новых примитивных функций преобразования неэквивалентных ТД для новых ЯП (C, C++, Python, Basic, Ruby, JAVA и др.) и других типов (рис.7).

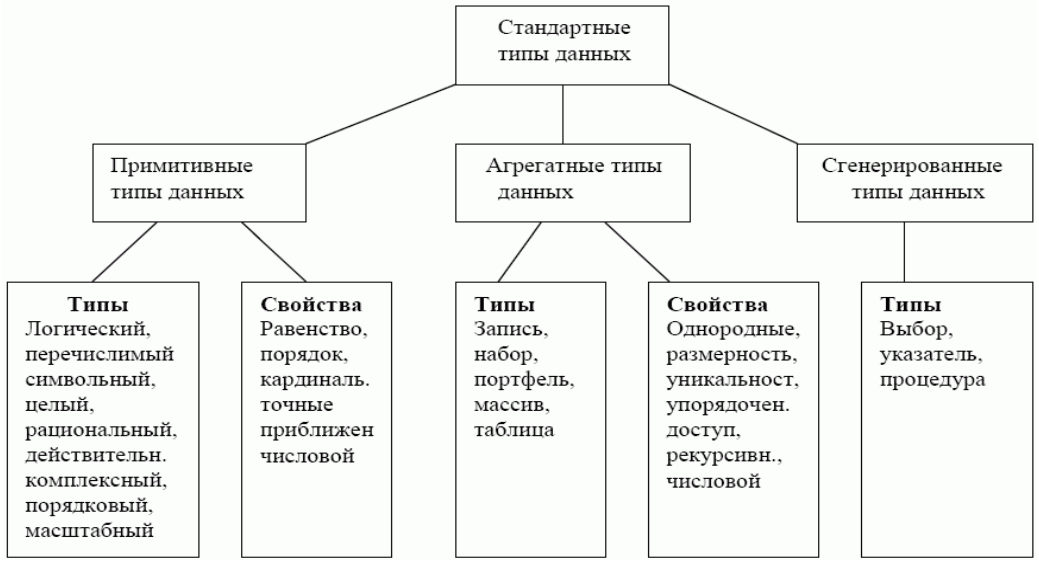


Рис. 7. Типы данных стандарта GDT

Согласно приведенной схемы реализуются следующие задачи

- преобразования данных для поддержки сборщика ресурсов:
- специфицирование внешних ТД в WSDL, сохранения их в БД и в репозиториях;
- преобразование новых ТД в новых ЯП1, …, ЯП n ;
- реализацию ТД GDT к виду специальных примитивных функций FDT;
- представление ТД FDT к виду специальных примитивных функций и формату Фреймворка;
- эквивалентные отображения и генерацию данных GDT<=>FDT с учетом платформ современных компьютерных и кластерных систем Интернет, где ресурсы будут работать.

Для решения проблем преобразования данных согласно стандарта GDT и FDT при связывании (взаимодействии) КПИ и сервисов, заданных в разных современных ЯП, предложен подход к генерации GDT<=>FDT (рис.8), Основу этого подхода составляет задачи преобразования типов данных в сборщике ресурсов с использованием ранее созданных для FDT .

Полуструктурированные, неструктурированные и расширяемые ТД данного стандарта используются в информационной среде, в связи с исследованиями недр земли, космоса и океана. Потребуется практическая обработка этих новых структур данных, чтобы с ними решались эффективно прикладные задачи в разных прикладных областях, особенно работающих с Big Data,

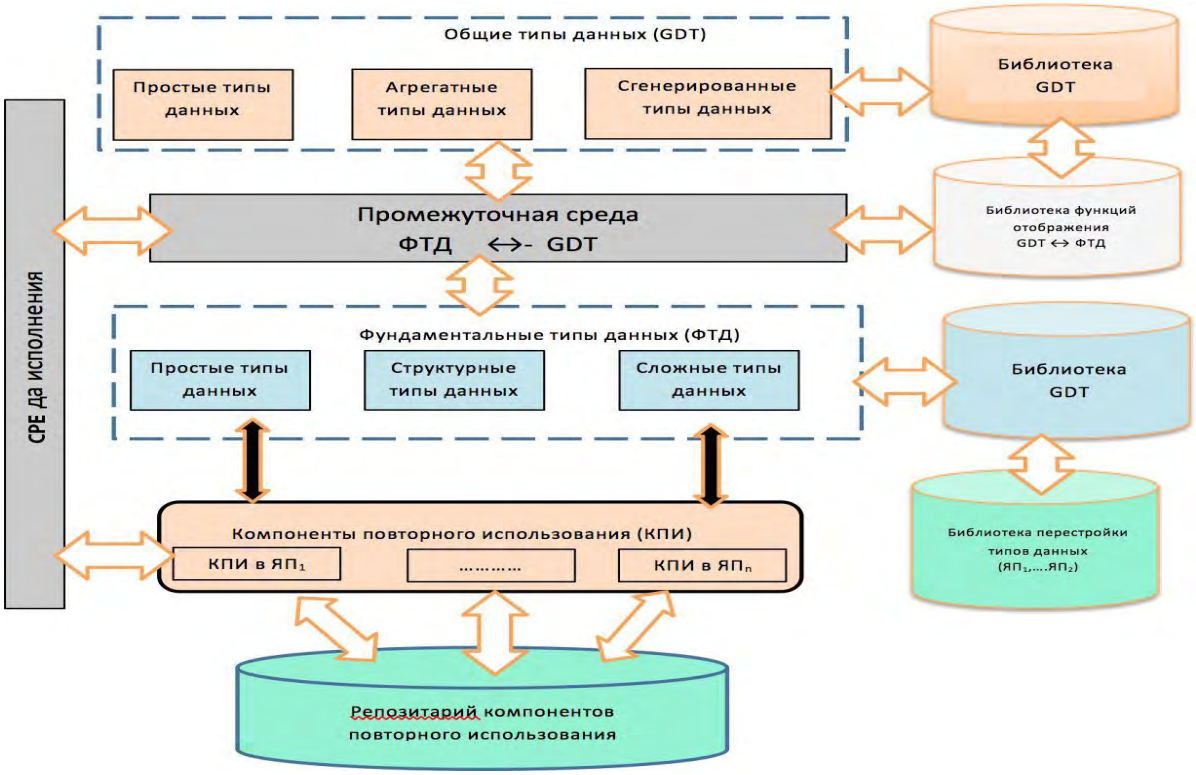


Рис.8. Схема преобразования ТД стандарта GDT

Подходы к обработке неструктурированных данных Big Data

Big Data — набор данных большого объема, а также подходов, средств, инструментов и методов представления неструктурированных огромных объёмов данных для получения данных и эффективного использования их на многочисленных узлах сети Интернет при решении прикладных Intelligence ИС и ИТ систем с использованием СУБД [16]. Для работы с большими объемами данных сформировался метод ETL (Extract Transform Load), с помощью которого производится:

- извлечение данных из внешних источников;
- трансформация и очистка данных с учетом требований;
- загрузка данных в хранилища данных;
- анализ данных и их перенос из одного приложения в другое и др..

К основным свойствам Big Data относятся:

- горизонтальная масштабируемость обрабатываемых больших данных и большого количества кластеров и серверов;
- отказоустойчивость по отношению к сбоям на процессорах кластеров и в узлах сети. Так, инструмент Hadoop-кластер Yahoo имеет более 42000 компьютеров, среди которых часть из них может выходить из строя;
- локализация данных и обработка их на серверах, где практически хранятся большие данные для решения соответствующих задач;
- изменение числа работающих на кластере с помощью средств MySQL Cluster. Большие данные могут быть представлены как tensors, которые управляют вычислением (например, полилинейное обучение подпространств Multilinear Subspace Learning и др.).

Системные средства обработки Big Data

NoSQL-БД нереляционные и распределенные данные с открытым кодом и горизонтальной масштабируемостью, эффективно поддерживают случайное чтение, запись и версионность.

MapReduce — модель распределённых вычислений, которая используется при параллельных вычислениях с большими данными в компьютерных кластерах.

Hadoop — свободно распространяемый набор утилит, библиотек и фреймворков для разработки и выполнения распределённых приложений (в том числе, MapReduce-программ), работающих на кластерах с сотнями и тысячами узлов.

СМБ — системы обработки больших данных в рамках Cloud-вычислений.

Big Data анализируются средствами: статистических и динамических методов анализа искусственного интеллекта, нейронных сетей, математической лингвистика; A/B Testing, Crowdsourcing Data Fusion; Integration Genetic Algorithms Machine Learning; Natural Language Processing; Signal Processing Simulation and Visualization; Massively Parallel Processing; Search-Based Applications, Data Mining, Multilinear Subspace Learning и др.

2.3. Технологии программирования (ТП) и путь ее развития после 1992

Теория ТП, как математическая дисциплина, сформулирована А.П. Ершовым и зарубежными учеными (E.Dijksra C. Hoar, Z. Manna, D.Gries, D.Buirner, K. Parnas и др.) [1-10]. Эта теория основывается на теории алгоритмов, множества, алгебры, логики исчислении предикатов, комбинаторики и др. В докладе на звание академика СССР (1984) и на Всесоюзной конференции «Технология программирования» (1986) [21-23] А.П.Ершов определил теорию ТП, которая включает методы синтеза, сборки и конкретизации. Синтезирующее программирование базируется на методе доказательного рассуждений о правильности алгоритма программы. Сборочное программирование осуществляет объединение совокупности существующих (проверенных на правильность) готовых фрагментов программ (reuses) для задач предметной области и сборку их в сложную структуру. Конкретизирующее программирование обеспечивает построение системы по универсальной модели для некоторой предметной области». Основные положения теории ТП он сформулировал в [22], как «конкретный способ организации, создания, распространения и сопровождения программного продукта (ПП).

Технология и методология – это наука, а метод входит в них составной частью. Технология начинается тогда, когда она охватывает ЖЦ создания ПП для предметной области.

Математическая наука программирования по мнению Ершова А.П. имеет три перспективных направления развития:

Первое направление (организационное программирование) 1975–1985 гг.

Второе направление (сборочное программирование) 1985–1995 гг.

Третье направление (доказательное программирование) 1995–2005 гг.

К методам ТП Ершов относил функциональное (В.П.Турчин), расслоенное А.Л.Фуксмана, расширяемое программирование М.М.Горбунов-Посадова и параллельное программирование [24-26] и др. После 1992года многие отечественные методы программирования слабо развивались и в основном применялись зарубежные технологии ООП, UML, VDM, Agile и др.

SEMAT (Software Engineering Methods and Theory (2009). Согласно стандарту SEMAT развитие теории программирования требуют математизации программирования для разных видов прикладных областей. (биология, генетика, медицина и др.) с помощью ЯП (C++, Java, Python, Ruby и др.) с объектной ориентацией и строгое математическое определение разных видов создаваемых прикладных систем с обеспечением качества.

В XXI веке согласно статьи B. Boehm “Perspectives and problems of Software Engineering” (IEEE Computer Society, v.41, N3 2008) определены проблемные задачи на ближайшие 10 лет:

- обеспечение изменений Legacy-систем (ОС IBM, ОС MS и др.) с применением средств dependability по модели CMM и масштабирование изменений;
- высокие цены на многомиллионное ПО компьютеров, базируется на методе сборки приложений из готовых платных ресурсов для разных областей;

- моделирование физических экспериментов проводить по методу ASC (Advanced Simulation Alliance) в суперкомпьютерных центрах;

- аутентификация серверов и защита каналов между клиентом и сервером (SSL и TSL) с открытыми ключами и паролями в перспективе требует управление ресурсами QOS (quality-of-service) с гарантией качества;

- развитие перспективных технологий средствами Grid-системы (рис.5).- http://www-fp.grids-center.org/news/pdf/HPDC13-Grid3)

Появились онтологические и интеллектуальные средства в Семантик Веб Интернет и проведено конвейерное производство ПП (APP.FAB) - фабрики программ (WebShpereIBM, MS.Net) разного назначения. Определилась концепция моделирования вариантов ОС (Linux, OS IBM, OS MS, Unix и др.) в рамках научного проекта VAMOS (Variability Model OS) и SEPL. Проведены конференции (EuroSys'11, SPLC'12, PlOS'12 and so on) для создания динамических вариантов систем ПО и ОС с помощью модели Kconfig.

Аспекты теории сборки и преобразования данных ресурсов

Сущность теории решения задачи сборки пар разнородных модулей или объектов состоит в построении взаимно однозначного соответствия между множеством формальных и фактических параметров [1].

V = {v1, v2, ..., vк} вызов объекта с множеством формальных параметров;

F = {f 1, f 2, ..., f к1} вызов объекта и их отображение с помощью алгебраических систем [2–4].

Каждому типу данных Tαt языка lα ставится в соответствие алгебраическая система вида:

Gαt = <Xαt , Gαt > ,

где Xαt — множество значений рассматриваемого типа данных (ТД), а Gαt — множество операций над объектами данного типа. Операциям преобразования ТД соответствует изоморфное отображение алгебраических систем Gαt в Gβd.

В классе алгебраических систем Σ = {G αb , Gαc , Gαi , Gαr , Gαa , Gαz}

где тип t – это b — boolean, c — character, i — integer, r — real, a — array, z — record и др. В системе Σ реализованы все виды преобразований для представленных ТД и проведено доказательство изоморфизма алгебраических систем [3] и его отсутствие для множеств неэквивалентных значений Xαt и Xβq. Установлено, что мощности алгебраических систем равны

|Gαt| = |Gβq|.

В парадигме сборки реализованы универсальные формальные основы преобразования типов входных / выходных данных (простых и сложных) FDT к общим типам данных GDT стандарта ISO/IC 11404 (примитивные, агрегатные, генерированные и неструктурированные) [1–10]. Ниже рассматриваются общие вопросы преобразования FDT, GDT и GDT<=>FDT, как очень важные для сборщика программ для среды Интернет.

Подходы к обработке неструктурированных данных Big Data

Big Data — набор данных большого объема, а также подходов, средств, инструментов и методов представления неструктурированных огромных объёмов данных для получения данных и эффективного использования их на многочисленных узлах сети Интернет при решении прикладных Intelligence ИС и ИТ систем с использованием СУБД [16].

Для работы с большими объемами данных сформировался метод ETL (Extract Transform Load), с помощью которого производится:

извлечение данных из внешних источников;

трансформация и очистка данных с учетом требований;

загрузка данных в хранилища данных;

анализ данных и их перенос из одного приложения в другое и др..

К основным свойствам Big Data относятся:

горизонтальная масштабируемость обрабатываемых больших данных и большого количества кластеров и серверов;
отказоустойчивость по отношению к сбоям на процессорах кластеров и в узлах сети. Так, инструмент Hadoop-кластер Yahoo имеет более 42000 компьютеров, среди которых часть из них может выходить из строя;
локализация данных и обработка их на серверах, где практически хранятся большие данные для решения соответствующих задач;
изменение числа работающих на кластере с помощью средств MySQL Cluster. Большие данные могут быть представлены как tensors, которые управляют вычислением (например, полилинейное обучение подпространств Multilinear Subspace Learning и др.).
К системным средствам обработки Big Data относятся:

NoSQL-БД нереляционные и распределенные данные с открытым кодом и горизонтальной масштабируемостью, эффективно поддерживают случайное чтение, запись и версионность.

MapReduce — модель распределённых вычислений, которая используется при параллельных вычислениях с большими данными в компьютерных кластерах.

Hadoop — свободно распространяемый набор утилит, библиотек и фреймворков для разработки и выполнения распределённых приложений (в том числе, MapReduce-программ), работающих на кластерах с сотнями и тысячами узлов.

CMD— системы обработки больших данных в рамках Cloud-вычислений.

Big Data анализируются средствами: статистических и динамических методов анализа искусственного интеллекта, нейронных сетей, математической лингвистика; A/B Testing, Crowdsourcing Data Fusion; Integration Genetic Algorithms Machine Learning; Natural Language Processing; Signal Processing Simulation and Visualization; Massively Parallel Processing; Search-Based Applications, Data Mining, Multilinear Subspace Learning и др.

2.3. Описание задач сборщика интеллектуальных и информационных ресурсов в Интернет

В качестве общего вывода по рассмотрению операций сборки можно сделать вывод о том, что приведенные операции сборки (link, make, config, assembling, building, weaver, integration) ресурсов в системных средах (3.2 -3.8), являются базовыми средствами для создания общего «сборщика» готовых ресурсов для прикладных, сервисных, информационных и технических систем в сетевой Глобальной сети Интернет.

Приведенные операции сборки (link, make, config, assembling, weaver) ресурсов в заданных системных средах в п.3.1-3.8 имеют схожие операционные способы сборки разных вариантов ресурсов: модулей в разных ЯП, исходных и выходных текстов компиляторных программ, сервисных и системных ресурсов Веб среды Интернет, технических средств компьютеров имеют похожие способы сборки, которые повторяются в разных общесистемных средах. Как бы они не назывались способы сборки семантика сборки остается одинаковой. Способ сборки зависит от данных, которые используются при обмене данными между связываемыми разнородными ресурсами.

Рассмотренные средства генерации общих типов данных стандарта ISO/IEC 11404 GTB – 2012 требуют создания набора новых примитивных функций для нестандартных типов данных (портфелей, контейнеров, шаблонов, указателей, неструктурных данных и др.) и использоваться в названных способах сборки разнородных ресурсов Интернет.

Для создания общего «сборщика» готовых ресурсов в прикладные и технические (макроконвейерные) системы в Интернет, необходимо на базе конфигурационной сборки сделать следующее:
Определить формальный вариант задания операции сборки ресурсов

Создать библиотеку примитивных функций для всех систем Интернет и дорабатывать примитивные функции для новые ТД (контейнеров, шаблонов и др.) согласно стандарта GDT.

Создать анализатор (агент) операции сборщика и взаимодействия (компиляторных, системных, прикладных, технических) с учетом всех типов ресурсов.

Проводить анализ соответствия типа ресурса и взаимодействия с другим ресурсом через интерфейс и осуществлять обращение к соответствующим программам преобразования обмениваемых данных для генерации соответствующего преобразования по теории преобразования типов данных стандарта GDT.

Управлять конфигурационной сборкой выполняя все процессы, определенные в стандарте IEEE 828 Configuration.

Выдавать сведения о результатах сборки и завершения работы сборщика.
Обеспечить размещение ресурсов в библиотеках и хранилищах Интернет из разных предметных областей знаний (медицины, биологии, генетики, математики и др.).

Техническим результатом сборки общего сборщика является:
простота поиска ресурсов в хранилищах Интернет и снижение затрат на разработку за счет готовых правильных многоразовых ресурсов и настройку их на конкретные условия применения; повышение качества и производительности создаваемых из ресурсов Веб-приложений и систем за счет системных операций замены отдельных более правильных ресурсов и изменения конфигурации (архитектуры) варианта конфигурационного файла с получением качественных решений функциональных задач приложений в разных предметных областей знаний.

2.4. Задачи сборка экспериментального варианта ядра OS Linux

OS Linux содержит более 10 000 переменных и большое множество функциональных системных компонентов, обеспечивающих обработку разного рода заданий по функционированию любых прикладных систем под управлением OS Linux представлен в отчете ЦИТИС-2019 и в схеме (рис.9 [34, 46]. Проведена Козин и Мутилиным и др.

- Лаврищева Е.М., Мутилтин В.С., Козин В..С. Рыжов А.Г. Моделирование прикладных и информационных систем из готовых ресурсов Интернет.- Труды ИспРАН, Том31, вып.1., 2019, стр.7-24
- Козин В.С. Конфигурационная сборка ядра OS Linux для прикладных систем.- Труды ИСП РАН,
  2018, том 31.-№ 5.-с. 73-80.

На верхнем уровне находится пользовательское пространство, где находятся приложения пользователей и библиотека GNU C (glibc) с интерфейсами системных вызовов для связи с ядром. Ядро OS Linux содержит более 11000 программных элементов и является общим для всех процессорных архитектур, поддерживаемых ОС Linux. Следующий уровень - архитектурно-зависимый код ядра BSP (Board Support Package) определяет конкретную архитектуру процессора и платформы. OS Linux можно откомпилировать для огромного количества разных процессоров и платформ, имеющих разные архитектурные ограничения и потребности. OS Linux может работать на процессоре как с блоком управления памятью (MMU), так и без MMU.

К основным компонентам ядра OS Linux относятся следующие:

- интерфейс системных вызовов функций ядра;
- управление процессами или потоками с помощью интерфейса программирования приложений (API) через SCI для создания нового процесса и алгоритма планировщика, время работы которого не зависит от числа потоков;
- управление памятью с участием аппаратных средств, устанавливающих соответствие между физической и виртуальной памятью;

- определение виртуальной файловой системой (VFS), которая предоставляет общую абстракцию интерфейса к файловым системам и служит для коммутации между SCI и файловыми системами ядра;

- сетевой стек имеет многоуровневую структуру самих протоколов.

Internet Protocol (IP) - это базовый протокол сетевого уровня, располагающийся ниже транспортного протокола Transmission Control Protocol, TCP). Выше TCP находится уровень сокетов, вызываемый через SCI. Уровень сокетов представляет собой стандартный API к сетевой подсистеме. Он предоставляет пользовательский интерфейс к различным сетевым протоколам;

- драйверы устройств обеспечивают возможность работы с конкретными аппаратными устройствами.

В OS Linux разработан гипервизор, с помощью которого можно строить вариант ОС для других систем. На основе ядра ОС выбраны необходимые компоненты и создана модель MF с базовыми характеристическими компонентов ОС и модели системы Msys, включая множество функциональных Mf,, интерфейсных Mio и Md работы с данными базового ядра OS Linux.

3. Технология сборки интеллектуальных и информационных ресурсов Интернет

Подходы к интеллектуализации систем и ресурсов

WWW и Семантик Веб Интернет ориентированы на создание интеллектуальных систем [8, 12, 17, 18]. В них процесс управления научными знаниями состоит в:

- отображении знаний в Базах Знаний и давать ими пользоваться другим;
- моделировании (knowledge modelling) знаний для их использования при решении конкретных прикладных задач;
- поиске и выборе (knowledge retrieval) знаний из различных хранилищ в подмножество контента для релевантного решения конкретной задачи;
- повторном использовании (knowledge reuse) знаний в виде готовых описаний, образцов (patterns), моделей в разнообразных контекстах;

публикации (knowledge publishing) знаний в стандартизованной форме для последующего распространения;

обновлении (knowledge maintenance) и сохранении знаний в Базах Знаний;

приобретении (knowledge acquisition) знаний для анализа неструктурированной информации (тексты, изображения, сценарии и др.), преобразования неявных знаний в явные, решения задач обработки данных огромных объемов и сохранения их в Базы Знаний.

Аспектом интеллектуализации информационных ресурсов (документов, текстов, таблиц, страниц и др.) являются процессы:

форматирование данных больших объемов в БД глобального масштаба с обеспечением сохранности, защищенности и безопасности;

поиска, выбора фрагментов данных и их транспортирование по запросам пользователей;

анализа, интеграции, агрегации информации (например, документов, в требуемые структуры для обработки на разных уровнях управления организациями страны и др.) для проведения вычислений задач и др.

2.4. Интеллектуализация знаний о ресурсах

К современным средствам представления знаний в терминах дескриптивной логики относятся: FaCT (на LISP), FaCT++ (на C++), CEL, KAON-2 (JAVA), MSPASS (C) и Pellet (JAVA) и др. Необходимой составляющей представления знаний являются прикладные web-сервисы, ориентированные на коллективное решение задач в предметных областях знаний в современных средах (problem-solving environments, PSE) в мировом сообществе.

В процессе создания моделей знаний предметных областей используются система KBS (Knowledge-based systems), CommonKADS и др. Они обеспечивают построение библиотек знаний, содержащих элементы решения задач, которые затем могут повторно использоваться и в других областях знаний.

Исходя из того, что программа на ЯП – это некоторый формальный или математический текст, то предложена система управления текстовой информацией и терминологией (System Quirk – SQ, http://www.computing.surrey.ac.uk/SystemSQ. Эта система ориентирована на создание и ведение терминов в терминологической базе данных (ТБД) и баз знаний (БЗ), а также организацию коллекций текстов на компьютере с помощью инструментов Virtual Corpus, KonText, Ferret, Grid и др.

Для ведения больших объемов данных и обеспечения функционирования систем в глобальной сети Интернет используются системные, сервисные и коммуникационные ресурсы, включая средства Семантик Веб. Семантик-Web — эта информационная среда в World Wide Web, которая предоставляет семантические ресурсы, языки, средства и инструменты разработки онтологий предметных областей знаний, прикладных систем и бизнес-процессов с использованием накопленных знаний в среде http://semwebprogramming.org.

В этой среде проводится автоматизированная обработка научных задач, больших данных в разных форматах, интеграция данных из коллажей (Mash-Ups), поиск и композирование веб-сервисов, управление интеллектуальными агентами в мобильных приложениях и др.

Одним из инструментов интеллектуализации знаний в Семантик Веб о разных предметных знаниях (математики, физики, биологии, медицины и др.) является онтология (www.semantic.web/ontology).

В основе онтологии представления знаний о некоторой области знаний лежит понятийная база, совокупность концептов (понятий) и отношений между ними, классификация понятий и их таксономия в виде тезаурусов, а также методы создания профильных онтологий. Онтология является инструментом построения концептуальной модели для областей знаний/доменов, которая включает объекты, классы объектов, структуры данных, связи и отношения (теоремы, ограничения) применительно к конкретной области знаний.

Имеется опыт разработки онтологии домена SE - ЖЦ ISO/IEC 12207 -2007 в языке OWL(Web Ontology Language) в среде Protégé 2.3 с выходным результатом в XML. Подход к автоматизации ЖЦ докладывался на Международной конференции «Science and Information-2015» (www.conference.thesai.org) в Лондоне и на XVII Всероссийской научной конференции-2015, 2016 «Научный сервис в сети Интернет» в г. Новороссийск, ИПМ им. М.В. Келдыша [18-20, 29, 33]. Таким образом, Семантик Веб предоставляет не только средства создания онтологий для предметных областей знаний, но и способ решения отдельных задач с использованием многочисленных сервисов и научных ресурсов по разным предметным областям в Интернет, в том числе и по бизнес-услугам. Для применения больших объемов информации при решении научных задач предметной области могут использоваться словари и концептуальные модели и приложений предметных областей знаний с помощью предметно-ориентированных языков (OWL, DSL и др.), инструментов FODA, Protégé, DSL Tool, KBuild, Kconfig (http://www.Sap.org) и др.

Интеллектуальные и информационные ресурсы

Технология сборки интеллектуальных и информационных ресурсов – это процессы сборки ресурсов (модульных элементов в разных ЯП), структур данных, процедур, классов, компонент и сервисов.

Ресурсы разрабатываются для разных прикладных областей знаний (математика, медицина, биология, генетика и др.). Они накоплены в библиотеках, хранилищах Интернет и в общесистемных средах (IBM, MS.Net, Unux, Intel и др.).

К хранилищам относятся библиотеки, репозитории, базы данных (БД). Библиотеки процедур, Базы знаний, Reusability Libraries и др. Функциональные элементы типа reuses создают высокоинтеллектуальные специалисты в области информатики, математики, биологии и др. (например, MatLab, Demral и др.). Каждый готовый reuse при размещении в библиотеки общего использования проверяется на правильность и качество реализации функции.

Интеллектуальные ресурсы (ИНР) – это компоненты повторного использования (КПИ и reuses), отображающие знания специалистов в разных областях знаний. Любая функция ПрО описывается в ЯП в виде модуля (объекта, компонента, сервиса и др.) и может взаимодействовать через link с другими модулями через операции типа CALL/RPC/RMI в текстах программ, в параметрах которых задаются данные для обмена между модулями, в IDL, API, ABI, WSDL и др. КПИ прикладных задач включают информационные данные, которые относятся к фундаментальным (FDT) и общим типам данных (GPD) стандарта ISO/IEC 11404, а также могут оперировать с большими объемами данных (Big Data). ИНР взаимодействуют с модулями, объектами или сервис-компонентами Интернет через интерфейс в языках IDL, API, ABI, WSDL и др.[16, 32, 33].

Reuse — это готовый интеллектуальный программный объект, компонент, сервис, реализующий алгоритм функции в некоторой предметной области знаний. Он специфицируется в стандартном языке WSDL и может многократно использоваться, если удовлетворяет функциональным требованиям отдельных предметных областей. КПИ как Reuses имеет код (implementation) с интерфейсом (interface) и схемой развертки (deployment) для выполнения. КПИ, Reuses могут иметь общие переменные, которые образуют классы или множества. Внешние общие переменные и методы класса задаются в интерфейсе экземпляров класса.
Информационные ресурсы (ИР) – это данные разного объема, в том числе и Big Data, представлены в Базах Данных, файлах, каталогах, таблицах, документах и т.п. ИР обрабатываются сервисными, системными, клиентскими и серверными службами Интернет и помещаются в Хранилищах данных, БД малых и больших объемов со структурированными и неструктурированными данными [17–21].

Интерфейс — это спецификатор информационной части ИНР — КПИ, компонента, reuses для обращения к другим ресурсам и обмена данными между ними. Интерфейс содержит вызов метода или функции (RPC/RMI) с параметрами, которые управляют внешними переменными экземпляра класса (выборка значения get-метод, присвоение значения set-метод, Home-интерфейс в языке JAVA и др.) при их взаимодействии [24, 25]. Интерфейс описывается в языке WSDL и в общем случае содержит:
название функции (метода) и ID — идентификатор ресурса;
описание (спецификацию) функции средствами ЯП;
параметры (входные и выходные) для передачи данных другим КПИ;
язык описания КПИ (C, C++, Java, Python, Ruby и др.);
необязательные атрибуты (дата, состояние, версия, право доступа, автор, срок использования и т.п.).

Среда программирования Eclipse позволяет автоматически создавать описания ресурсов на основе классов языка JAVA. В этом языке определены следующие основные типы данных:
строки (xsd:string);
целые числа (xsd:int, xsd:long, xsd:short, xsd:integer, xsd:decimal), числа с плавающей запятой (xsd:float, xsd:double);
логический тип (xsd:boolean);
последовательности байт (xsd:base64Binary, xsd:hexBinary);
дата и время (xsd:time, xsd:date, xsd:g);
объекты (xsd:anySimpleType).

В качестве переменных могут использоваться множества, последовательности, включающие фиксированное количество переменных простых типов.
Типичный WSDL-файл имеет следующую структуру.

```
<wsdl:definitions [.]>
<!-- Декларация типов, которые используются в сервисе -->
<wsdl:types>
<element name="someMethod">
<complexType>
<sequence>
<element name="arg0" type="xsd:double"/>
<element name="arg1" type="xsd:boolean"/>
</sequence>
</complexType>
</element>
<element name="someMethodResponse">
<complexType>
</wsdl: types> …
```

Приведенное WSDL-описание задает веб-сервис MyService с единственным методом String someMethod (double arg0, boolean arg1). На его основе можно сгенерировать два типа данных, которые соответствуют входным и выходным параметрам метода. Эти типы применяются в описаниях someMethodRequest и someMethodResponse — входного и выходного сообщений для операции someMethod.
Операции декларируются в описании интерфейса сервиса (декларация wsdl:portType) и затем дается описание привязки сервиса к протоколу SOAP (Simple Object Access Protocol) через декларацию wsdl:binding. При этом устанавливается способ вызова <wsdlsoap:body use="literal"/> с параметрами, заданными в методе класса. В конце WSDL-файла приводится декларация веб-сервиса (<wsdl:service>), в которой содержится информация о расположении <параметр location>.

Сервисные и информационные ресурсы Интернет

Web-сервисы основаны на открытых стандартах Интернет, которые широко поддерживаются на платформах Unix, Intel, Windows, IBM, Linux и др. и задаются PHP, ASP, JSP-скрипт, JavaBeans и др.
Формат запросов к Web-сервисам определяет протокол SOAP, который задается в XML и отправляется через HTPP к Web-серверу. IBM, Microsoft и Universal Description, Discovery and Integration (UDDI) способствуют созданию общего каталога Web-сервисов. SOAP, XML и UDDI обеспечивают также надежность функционирования приложений из Web-сервисов и сервис-компонентов.
К средствам создания прикладных систем в Интернет относятся сервис-ориентированная архитектура SOA (Service Oriented Architecture) и сервисно-компонентная архитектура SCA (Service-Component Architecture).
SOA задает функциональность (Functions) и качество сервисов (Quality service) на уровнях:
транспортный (transport layer) для обмена внешними данными на коммуникационным уровне (service communication layer);
описания сервиса и интерфейса (service description layer) с обеспечением безопасности и защиты (авторизации, аутентификации);
операций публикации, поиска и вызова сервисов через реестр сервисов.
В Интернет имеется реестр сервисов (рис.1), который содержит описание сервисов в языках Семантик Веб (SOA, SCA, SMC и др.) и обеспечивает регистрацию, поиск и вызов сервиса по запросам поставщиков и потребителей сервисов [12, 28, 30].

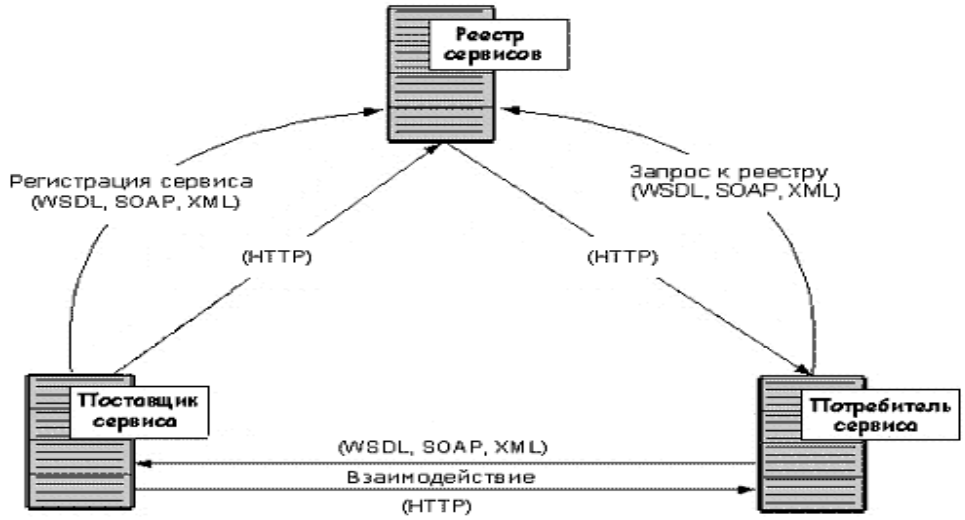


Рис. 1. Служба сервисов в сетевой среде

Рассмотрим сервисы SOA и SCA. Операции над сервисами SOA такие:
публикация сервиса через вызов сервиса и его интерфейс;
поиск сервиса с помощью протокола SOAP;
связь UDDI с моделями CORBA, DBMS, JNET и т. п.;
запрос к провайдеру за опубликованными интерфейсами сервиса.

Сервис SCA дает доступ к сервис-компонентам, которые упаковываются в модуль выполнения сервиса в среде WebSphere, эквивалентного EAR-файлу J2EE. Сервисы SCA интегрируются через интерфейс JAVA и реализуются в виде классов JAVA™. В модели SCA объекты данных представлены в JAVA common.sdo.DataObject, который включает в себя метод определения свойств данных. WebSphere Integration Developer платформы Eclipse 3.0 позволяет композировать (собирать) сервисы SCA и сервисные интерфейсы.
Для вызова внешнего сервиса используется JMS, Enterprise JavaBeans или готовые сервисы. Доступ к БД системы проводится через аппарат ссылок. Веб-система собирается из сервисных компонентов, описанных в языках Python, JAVA, BPEL, OWL и интерфейсов с помощью операции конфигурирование Kconfig SAP [18, 19]. Библиотека SCA содержит набор reuses композитных сетевых сервис-компонентовх и гетерогенных данных (http://www.ibm.com/developerworks/websphere/techjournal).
Эти сервисы могут создаваться в среде Java Enterprise Edition [18, 19], которая позволяет проводить:
динамическую генерацию серверных страниц (Java Server Pages);
определение КПИ в виде Enterprise Java Beans;
преобразование параметров КПИ к формату представления других сред;
обмен сообщениями (Java Message Queue) со службами JMS (Java Message Service).
В SCA могут создаваться новые сервисные компоненты для проблемных областей знаний, а также объекты данных (Service Data Objects — SDO) и интерфейсы, обеспечивающие передачу данных другим КПИ. Для вызова внешнего сервиса используется JMS, Enterprise JavaBeans или готовые сервисы. Доступ к БД из системы (предприятия, фирмы) проводится через аппарат ссылок. Создаваемая Веб-система собирается из сервисных компонентов, описанных в языках Python, Java, BPEL, OWL и интерфейсов с помощью операцией конфигурирования Kconfig SAP (рис. 2).

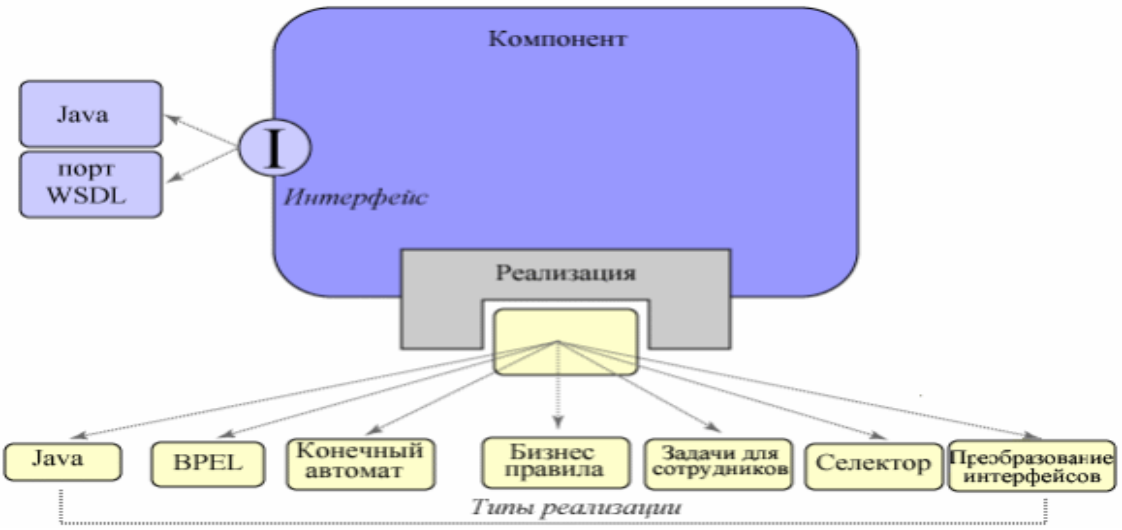


Рис.2. Общая схема сервиса SCA IBM

При работе веб-сервисов с даннями из класса Big Data используются такие средства Интернет: IBM WSDK+WebSphere; Apach Axis+Tomcat; Apach Axis+ Classfish; Microsoft.Net+IIS и др.

Системные ресурсы сети Интернет

К основным системным ресурсам Интернет относится клиент-серверная архитектура сети Интернет [37, 38], которая является трехуровневой: клиент, сервер приложений (систем) и сервер БД.

На уровень клиента выносятся простые сервисные ресурсы веб-систем: интерфейс, операции с данными, алгоритмы шифрования, взаимодействия и безопасности для передачи серверу приложений и т.п. Клиентская часть отвечает за интерфейс и обработку приложений Интернет, которые записаны на ЯП (C, C++, Python, Ruby, Java, Basic и др.) и выполняют прикладные функции, обработку возникающих аварийных ситуаций, контроль безопасности и качества и др.

Сервер БД запускается с сервера приложений и выполняет обслуживание БД с обеспечением целостности, сохранности данных при доступе клиента к информации БД. Для работы с Big Data выполняются задачи анализа данных большого объема, а также манипулирования данными с малой и большой нагрузкой.

Клиенту соответствует Интернет-браузер (Chrome, Firefox, MS Internet Explorer, Safari, Opera и др.). Он посылает сообщения серверу через интерфейс следующего вида: WebAppInterface = {Requestp}, где Requestp — p-й запрос.

Обработка запроса производится компонентами сервера вида: Internet Information Server, Apache, Tomcat, JBoss, WebSphere, WebLogic, Cloudscape.

Web-сервер Apache и JAVA обеспечивают обработку ИНР в современных ЯП и имеют вид: IR APACHE = {PERL, PHP, PYTHON, XML, SQL}, RJAVA = {JAVA, JSP, JSTL, JSF, XML, SQL} в классе системных компонентов (Tomcat, JBoss, WebSphere, WebLogic и др.).

Сервер Интернета включает средства: ASP, JavaScript, VB Script, ASP.NET, .NET, J#.NET, XML, SQL) и языки (C, C++, C#, Python, Ruby и др.).

ИР данных включает: модели данных, операции хранения и доступа к данным, обработку данных и др. Они выполняют доступ к данным и взаимодействие с ресурсами типа entity в модели EJB. При работе с большими объемами данных метод ETL (Extract Transform Load) переносит данные из одного приложения в другое через клиент-серверную архитектуру и проводит извлечение данных из внешних источников, их анализ и трансформацию к виду требований концептуальных моделей и выполнение в среде Интернет.

5.10.. Технология сборки сервисных ресурсов в Интернет

Основу технологии составляет метод сборки ресурсов и интерфейс связи разнородных КПИ в архитектуру с взаимно однозначным преобразованием типов входных и выходных данных. Межмодульный интерфейс — совокупность средств формального преобразования разноязыковых КПИ. Интерфейс между КПИ включает набор формальных средств организации обмена данными между модульными элементами [26, 33]. Метод сборки основан на математических формализмах спецификации связей (по данным и по управлению) между разнородными модульными элементами и генерации интерфейсных модулей-посредников для каждой пары связанных элементов. Каждая связь задается оператором вызова CALL/RPC/RMI в ЯП с набором фактических и формальных параметров [5].

Технология сборки ресурсов основана на определении:

– схем, графов и моделей предметной области знаний;
– репозитория модулей (КПИ, Reuses, процедур и др.);
– алгоритмов модулей с операциями CALL/RPC /RMI вызова модулей;
– интерфейсных модулей с параметрами передаваемых данных;
– операций хранения и выбора модулей, КПИ из репозиториев;
– верификации, тестирования модулей и их интерфейсов и др.;
– функциональной надежности и качества ресурсов и систем из них.

В практике программирования сформировались системы сборки модулей с помощью операций сборки: link АПРОП (CORBA); make компиляторных программ BSD, GNU, MSBuild; assemble, Kconfig системных, сервисных и вычислительных ресурсов EuroGrid; integration config сервисных SOA, SCA ресурсов в Semantic Web; weaver BEA WebLogic Oracle, SAP Net.
Способы сборки описаны в данной книге на стр. 205-209. По ним сделана общая публикация с Петренко А.К. И Позином. На конференции Абрау 19 и сделан на англ. языке.
В разделе 2 и 21 описаны метод сборки информационных и интеллектуальных ресурсов и идея общего сборщика этих ресурсов в Интернет. Далее рассматривается индустрия

Индустриальные системы сборки ресурсов

Способ сборки по типу фабрик программ в IBM WebSphere, Microsoft Biz Talk, BEA WebLogic Oracle 10g, SAP NetWeaver и ИВК «Юпитер». В них содержатся CASE средства сборки (link) ресурсов (сервисов, компонентов и данных через брокера обмена данными stub и skeleton системы CORBA, переедаемых данные с помощью протокола Workflow с проверкой безопасности и качества (табл. 2) [24].

Таблица 2. CASE–системы сборки ресурсов

Платформа	Фирмы	Содержание платформы
IBM WebSphere	Корпорация IBM	Сервер приложений J2EE, брокеры обмена данными, КПИ, портал, workflow/BPM, EII, SCA
Microsoft Biz Talk 2004 и компоненты .Net	Корпорация "Майкрософт"	Сервер приложений COM, брокеры обмена данными, RGB доставка, портал, workflow/BPMN
BEA WebLogic	Корпорация "BEA Systems" (с 2008г. входит в "Oracle")	Сервер приложений J2EE, брокеры обмена данными, ГОР, сервер прикладных приложений, портал, workflow/BPMN
Oracle 10g	Корпорация "Oracle" http://www.oracle.com	Сервер приложений J2EE, брокеры обмена данными, ГОР, портал, workflow/BPMN, средства EII
SAP NetWeaver	Корпорация SAP http://www1.sap.com/www.sap.ru	Сервер приложений J2EE/ABAP, брокеры обмена данными, портал, инструменты BPMN

ИВК "Юпитер	Компания ИВК (Россия) http://www.ivk.ru/	Брокеры обмена данными, КПИ, среда выполнения, сертификация, защита данных

На фабриках программ этих систем модифицируется архитектура в соответствии с требуемыми кодами сервисных ресурсов модели SOA в Visual Studio Industry Partners (VSIP). При этом используются модели Guidance Automation eXtensions (GAX) и Guidance Automation Toolkit (GAT) в Visual Studio. GAX — это среда исполнения в VSIP с использованием пакетов рекомендаций. Существует два варианта служб: веб-служба ASP.NET (ASMX) для Windows Communication Foundation (WCF) в среде .NET Framework 3.0. Версии для веб-службы WCF находятся у разработчиков фабрики ПО GotDotN. В ее состав входят процессы предприятия и пакеты документаций и рекомендаций для приложения типа Global Bank. Это аналогично модели SCA в IBM WebSphere.

CASE инструменты автоматизации сборки программ

Способ сборки (интеграции) разных программ на CASE инструментах: Make, Apache Ant, Apache Maven, Gradle и др.

Make — это кросс-платформенная система автоматизации сборки систем из исходного кода. Она генерирует файлы управления сборкою, например Makefile в системах Unix для сборки посредством операции make (см. 3.4.)

Apache Ant — JAVA-утилита для автоматизации процесса сборки программного продукта. Ant — платформо-независимый аналог UNIX-утилиты Make, но с использованием языка JAVA, приспособленного для JAVA-проектов. Самая важная разница между Ant и Make состоит в том, что Ant использует язык XML для описания процесса сборки и его зависимостей, а Make имеет свой собственный формат файл Makefile. XML-файл (или build.xml) осуществляет сборку исполняемых программ.

Apache Maven — программный инструмент для управления (management) JAVA-проектами методом сборки (building) ПО аналогично Apache Ant, но с более простой build-моделью конфигурации, которая базируется на формате XML. Двигатель ядра инструмента динамически загружает плагины из репозитория и обеспечивает доступ ко многим версиям разных JAVA-проектов с открытым кодом Apache.

Gradle — система автоматической сборки, построенная на принципах Apache Ant и Apache Maven, но вместо XML-подобной формы здесь используются языки DSL и Groovy для представления конфигурации создаваемого приложения или проекта.

Конфигурирование ресурсов в компонентной в среде GDM

Способ сборки ресурсов в индустрии. К. Чернецки создал генерационную мультисборку на ProductLine/ProductFamily. (см. Порождающее программирование. Методы, инструменты, применение,2005.-750с.). В модель GDM введены новые понятия для представления предметных областей знаний: пространство задач (problem space), пространство решений (solution space) и база конфигурации (configuration base) семейства систем (СПС). Пространство задач отображает понятия СПС, членов СПС и их общие характеристики в модели MF (Feature Model), а также функции и задачи, которые описываются GPL (General-Purpose Language) или предметно-ориентированными языками DSL, UML2. Пространство решений – это компоненты, каркасы, шаблоны и КПИ реализации задач членов семейства СПС.

Механизмы, правила описания, генерации компонентов и подбора КПИ для отдельных задач СПС входят в базу конфигурации. При этом каркас оснащается изменяемыми параметрами модели, что может привести к излишней его фрагментации и появлению "множества мелких методов и классов". Каркас обеспечивает динамическое связывание аспектов и компонентов в процессе реализации изменчивости между разными приложениями. Образцы проектирования обеспечивают создание многократно используемых решений в различных ПС. Для задания таких аспектов, как

синхронизация, удаленное взаимодействие, защита данных и т. д., применяются компонентные технологии ActiveX и JAVABeans, а также новые механизмы сборки
Результаты описания модели СПС в этих пространствах находятся в конфигурационной базе знаний (БЗ). Это связи и характеристики (функциональные или не функциональные), заданные в соответствующей модели MF членов семейства и выполняются операциями конструирования и объединения компонентов в общую ПС или СПС. Иными словами, в БЗ отображены знания о конфигурации системы в виде абстракций общего и специального назначения, КПИ и знания о новых компонентах, результатах их тестирования, измерения и оценивании.
Сборка по модели трансформация и конфигурация
Трансформационная модель приложения или домена описывается в языке DSL (из пространства проблем может превращаться в пространство решений путем трансформации DSL-спецификации модели в реализацию более простых ЯП .

Главный механизм перехода от описания приведенных моделей к исходному результату – трансформация описаний понятий домена в промежуточный язык DSL-пространства решений, а дальше в язык реализации компонентов с учетом платформы, где расположены готовые компоненты и/или новые задачи.
Пространство проблемы1 входят отдельные аспектов проблем. В зависимости от них трансформация может задаваться в языке реализации или другом DSL-языке (фактически, язык другой предметной области знаний).
Модель конфигурации базируется на конструкторских правилах, которые оптимизируют абстракции и характерные черты домена, Модель СПС формируется конфигурационный файл системы.

Описание специфики домена в DSL-языке трансформируется к ЯП компонентов пространства задач для последующей их генерации. При конфигурационном способе представления пространства проблемы с общими характеристиками и ограничениями фактически задаются понятия в проблемно-ориентированном языке и множество компонентов решений в ЯП.
Данными для конфигурационного способа преобразования пространства задач являются понятия предметной области, их характеристики (свойства) и недопустимые сочетания характеристик, согласованные по умолчанию параметров и зависимостей между этими элементами. Созданная из них конфугурационная модель реализуется по правилам конфигурирования и оптимизации в пространстве решений.
После отображения это пространство содержит множество компонентов (КПИ, reuse), схемы сборки готовых компонентов, варианты архитектур некоторых членов семейства.
Все эти элементы входят в конфигурационный файл пространства решений, а также в конфигурационную базу знаний.
При конфигурационном подходе может применяться другой язык описания архитектуры ADL (Arhitecture Description Languages).
По правилам описания архитектуры, трансформация описаний характеристик и ограничений MF производится в описание обобщенной архитектуры семейства ПС в языке ADL. Следующая трансформация описаний компонентов выполняется средствами ЯП. Конфигурационный файл содержит все элементы реализации компонентов и интерфейсов связи компонентов в IDL, а также для выполнения и внесения изменений в структуру ПС и СПС.

Подходы к обработке неструктурированных данных Big Data

Big Data — набор данных большого объема, а также подходов, средств, инструментов и методов представления неструктурированных огромных объёмов данных для получения данных и эффективного использования их на многочисленных узлах сети Интернет при решении прикладных Intelligence ИС и ИТ систем с использованием СУБД [16].

Для работы с большими объемами данных сформировался метод ETL (Extract Transform Load), с помощью которого производится:

- извлечение данных из внешних источников;
- трансформация и очистка данных с учетом требований;
- загрузка данных в хранилища данных;
- анализ данных и их перенос из одного приложения в другое и др..

К основным свойствам Big Data относятся**:**

- *горизонтальная масштабируемость* обрабатываемых больших данных и большого количества кластеров и серверов;
- *отказоустойчивость* по отношению к сбоям на процессорах кластеров и в узлах сети. Так, инструмент Hadoop-кластер Yahoo имеет более 42000 компьютеров, среди которых часть из них может выходить из строя;
- *локализация данных и о*бработка их на серверах, где практически хранятся большие данные для решения соответствующих задач;
- *изменение числа работающих* на кластере с помощью средств MySQL Cluster. Большие данные могут быть представлены как tensors, которые управляют вычислением (например, полилинейное обучение подпространств Multilinear Subspace Learning и др.).

К системным средствам обработки Big Data относятся:

NoSQL-БД нереляционные и распределенные данные с открытым кодом и горизонтальной масштабируемостью, эффективно поддерживают случайное чтение, запись и версионность.

MapReduce — модель распределённых вычислений, которая используется при параллельных вычислениях с большими данными в компьютерных кластерах.

Hadoop — свободно распространяемый набор утилит, библиотек и фреймворков для разработки и выполнения распределённых приложений (в том числе, MapReduce-программ), работающих на кластерах с сотнями и тысячами узлов.

CMB— системы обработки больших данных в рамках Cloud-вычислений.

Big Data анализируются средствами**:** статистических и динамических методов анализа искусственного интеллекта, нейронных сетей, математической лингвистики; A/B Testing, Crowdsourcing Data Fusion; Integration Genetic Algorithms Machine Learning; Natural Language Processing; Signal Processing Simulation and Visualization; Massively Parallel Processing; Search-Based Applications, Data Mining, Multilinear Subspace Learning и др.

Задачи общего сборщика ресурсов систем и индустрии фабрик программ

В качестве общего вывода по рассмотрению операций сборки для разных сред (3.1-3.7) можно сделать вывод о том, что приведенные операции сборки (link, make, config, assembling, building, weaver, integration) ресурсов в системных средах, являются базовыми средствами для создания общего «сборщика» готовых ресурсов для прикладных, сервисных, информационных и технических систем в Глобальной сети Интернет.

Приведенные операции сборки (link, make, config, assembling, weaver) ресурсов в заданных системных средах в п.3.1-3.8 имеют схожие операционные способы сборки разных вариантов ресурсов: модулей в разных ЯП, исходных и выходных текстов компиляторных программ, сервисных и системных ресурсов Веб среды Интернет, технических средств компьютеров имеют похожие способы сборки, которые повторяются в разных общесистемных средах. Как бы не назывались способы сборки семантика сборки остается одинаковой. Способ сборки зависит от данных, которые используются при обмене данными между связываемыми разнородными ресурсами.

Рассмотренные средства генерации общих типов данных стандарта ISO/IEC 11404 GPD – 2012 требуют создания набора новых примитивных функций для нестандартных типов данных

(портфелей, контейнеров, шаблонов, указателей, неструктурных данных и др.) и использоваться в названных способах сборки разнородных ресурсов Интернет.

Для создания общего «сборщика» готовых ресурсов в прикладные и технические (макроконвейерные) системы в Интернет, необходимо на базе конфигурационной сборки сделать следующее:

3. Определить формальный вариант задания операции сборки ресурсов
4. Создать библиотеку примитивных функций для всех систем Интернет и дорабатывать примитивные функции для новые ТД (контейнеров, шаблонов и др.) согласно стандарта GPD.
5. Создать анализатор (агент) операции сборщика и взаимодействия (компиляторных, системных, прикладных, технических) с учетом всех типов ресурсов.
6. Проводить анализ соответствия типа ресурса и взаимодействия с другим ресурсом через интерфейс и осуществлять обращение к соответствующим программам преобразования обмениваемых данных для генерации соответствующего преобразования по теории преобразования типов данных стандарта GPD.
7. Управлять конфигурационной сборкой при выполнии всех процессов, определенных в стандарте IEEE 828-2006 Configuration.
8. Выдавать сведения о результатах сборки и завершения работы сборщика.
9. Обеспечить размещение ресурсов в библиотеках и хранилищах Интернет из разных предметных областей знаний (медицины, биологии, генетики, математики и др.).

Техническим результатом общего сборщика является:

1) простота поиска ресурсов в хранилищах Интернет и снижение затрат на разработку за счет готовых правильных многоразовых ресурсов и настройку их на конкретные условия применения;

повышение качества и производительности создаваемых из ресурсов Веб-приложений и систем за счет системных операций замены отдельных более правильных ресурсов и изменения конфигурации (архитектуры) варианта конфигурационного файла с получением качественных решений функциональных задач приложений в разных предметных областей знанийВопросы обеспечения качества, безопасности и защиты данных см. в книге. **Лаврищева Е.М., Петренко А.К.. Кулямин В.В, Карпов Л.Е, Мутилин В.С., Козин С.В., Зеленов С.В., Рыжов А.Г. МОДЕЛИРОВАНИЕ И ПРОГРАММИРОВАНИЕ ВАРИАБЕЛЬНЫХ ПРИКЛАДНЫХ И ОПЕРАЦИОННЫХ СИСТЕМ С ОБЕСПЕЧЕНИЕМ БЕЗОПАСНОСТИ, НАДЕЖНОСТИ И ЗАЩИТЫ ДАННЫХ.М.ИСП РАН .-2021-2023.-321с.**

.

Продолжение табл. 1

C	

УДК 004.06

Е. М. Лаврищева, д-р физ.-мат. наук, проф., КНУ имени Тараса Шевченко, г. Киев, Украина, e-mail: Lavrischeva@gmail.com

Развитие идей академика В. М. Глушкова по технологии компьютеров, систем и программ

Статья посвящена 90-летию со дня рождения академика Академии наук Украины и СССР Виктора Михайловича Глушкова, она отражает его вклад в разработку технологии ЭВМ, компьютерных систем и программ. В 1970-х годах он предвидел, что появятся фабрики по созданию компьютеров, систем и программ, которые будут работать по принципу сборки, как в автомобильной промышленности на заводах Г. Форда. Со временем его предвидение оправдалось. Развивая концепцию сборки Глушкова, автор формализовала технологию сборки разнородных модулей в сложные программы и дала классификацию новых дисциплин, способствующих более наукоемкому и обоснованному созданию качественных программных продуктов для массового использования.

Ключевые слова: *технология программирования, технология компьютеров, систем и программ, фабрики программ, индустрия, методы и инструменты*

E. M. Lavrishcheva

Development of Academics Ideas of V. M. Gluchkov from the Technologies of Computers, Systems and Programs

An article is dedicated to 90-yeag of birth day of academician of NAN Ukraine and USSR Victor Michalovich Gluchkov, to his deposit in development of technology of ECM, computer systems and programs. In the 70-years of past century he foresaw, that the factories of computers, systems and programs, which will work on principle of assembling will appear, as in motor-car industrial factory of Ford. In course of time his foresight were justified. Offered them technologies are adequate to technologies, which appeared simultaneously in Computer Sciences. One of technologies — Software Engineering technology of programming is identical. Developing the Gluchkov's ideas, the author realized the technology of assembling ready components for the heterogeneous systems and gave a new classification of SE disciplines, oriented on industry of software products on the factory.

Keywords: *technology of programming, technology of computers, systems and programs, factories of the programs, industry, methods*

Академик Виктор Михайлович Глушков посвятил свою жизнь созданию отечественной кибернетической школы. Он оставил много парадигм, которые стали основой для формирования новых современных научных направлений, а также концептуальных положений, в значительной степени определивших последующее развитие кибернетики и информатики. Его ученики Ю. В. Капитонова и А. А. Летичевский к 80-летию В. М. Глушкова написали монографию [1], в которой выделили семь научных парадигм Глушкова: математическая теория проектирования ЭВМ; самоорганизация и совершенствование компьютерных систем; теория доказательства теорем, алгебраическое программирование для вычисления математических задач; принципы и методы построения информационных систем, автоматизированных систем управления (АСУ) и АСУ технологическими процессами (АСУ ТП); искусственный интеллект и концепция повышения внутреннего языка ЭВМ до уровня интеллекта человека; принципы диалогового взаимодействия человека с компьютерной и информационной средой; экономические модели и пути их совершенствования в системах управления.

К списку перечисленных парадигм относится и технология ЭВМ, систем и программ, которую академик В. М. Глушков определил в 1960-х гг., как повышение уровня программирования вычислительных задач на ЭВМ и позже как сборочный конвейерный

Академик Виктор Михайлович Глушков

способ постепенного перехода от ремесленного к промышленному производству компьютеров, систем и программ. Автор данной статьи работала непосредственно с В. М. Глушковым по проблематике технологии создания сложных систем и многие годы развивает его сборочный тезис. Технологию он считал двигателем прогрессивного развития любой науки, в том числе теории создания ЭВМ и программного обеспечения ЭВМ и автоматизированных систем (АСУ, АСУ ТП, САПР и др.).

Под руководством В. М. Глушкова в 1957 г. в Институте кибернетики Академии наук Украины начали создавать новые ЭВМ, теории и технологии, связанные с ними, в их числе:

- теория построения ЭВМ, вычислительных машин для инженерных расчетов "Мир", ЭВМ для управления технологическими процессами предприятий, машин со схемной интерпретацией языков программирования "Украина", интеллектуальных, мозгоподобных ("умных"), рекурсивных, макроконвейерных многопроцессорных машин и др.;
- методы автоматизации процессов создания системного программного обеспечения новых ЭВМ (операционные системы, трансляторы, редакторы и подобные им), больших систем, пакетов прикладных программ разного назначения (математического, экономического, транспортного и др.);
- теория и практика АСУ в целом и АСУ ТП на предприятиях СССР как общегосударственной сети управления предприятиями, оборудованными ЭВМ и устройствами сбора, обработки данных на автоматизированных рабочих местах (АРМ).

Виктор Михайлович Глушков рассматривал технологию создания ЭВМ как комплексное проектирование вычислительных систем, технических средств и их базового математического и системного обеспечения. Именно по такой технологии разрабатывалась серия машин "Мир" ("Мир-1", "Мир-2", "Мир-3") для инженерных расчетов с использованием математических методов, формульных алгебраических преобразований, математического доказательства теорем. В это время у В. М. Глушкова возникли концепция интеллектуальной "умной" (мозгоподобной) машинной структуры, частично промоделированной в проекте машины "Украина" [2—4] и позже в программно-техническом комплексе "Маяк" (1985—1991 гг.), а также концепция управляющих ЭВМ для АСУ и АСУ ТП, нашедшая отображение в управляющей машине "Днепр-1" (1962—1964 гг.) и в управляющем вычислительном комплексе (УВК) "Днепр-2" (1965—1969 гг.) [5, 6].

Технологии ЭВМ

При изготовлении первой ЭВМ под руководством академика С. А. Лебедева сформировалась технология проектирования и изготовления ***универсальных*** ЭВМ. Она совершенствовалась в плане унификации элементной базы и методов сборки отдельных элементов в более сложные структурные компоненты ЭВМ. По этой технологии изготавливались в СССР такие ЭВМ, как МЭСМ, Стрела, БЭСМ, М-20 и др. В совместной работе В. М. Глушкова и В. А. Мясникова были предложены новые виды рекурсивных, микро- и макроконвейерных ЭВМ с новой элементной базой для организации высокопроизводительных вычислений, решения сложных математических и народно - хозяйственных задач.

Принципы построения высокопродуктивной суперЭВМ макроконвейерного типа и реализации на ней семейства языков программирования для проектирования вычислительных систем В. М. Глушков

обосновал на конференции в Новосибирске (1979 г.) и опубликовал в журнале "Кибернетика", 1981, № 1. Эти концепции и принципы были реализованы (1985—1991 гг.) коллективом отдела В. М. Глушкова (А. А. Летичевский, Ю. В. Капитонова, Г. И. Горлачь, Н. М. Мищенко, С. С. Гороховский и др.) в системе "МАЯК" на макроконвейере ЭВМ ЕС 2701, который был принят Государственной комиссией СССР для задачи ядерного взрыва (1990 г.). На этих идеях построены высокопроизводительные многопроцессорные кластеры СКИТ-3 (2010 г.) и СКИТ-4 (2013 г.) для организации больших вычислительных задач и современный интеллектуальный комплекс "Инпар- ком" (1987—2013 г.).

Тезис конвейерной сборки сложных программ из готовых программных заготовок с использованием фондов алгоритмов и программ В. М. Глушков высказал на научном семинаре института (1975 г.). Этот тезис сначала получил развитие в системе АПРОП [7, 8], далее — в сборочном программировании и на линиях сборки компьютерных продуктов (машин и систем) на фабриках по производству компьютеров и программ для массового применения [9].

Базовые принципы построения управляющих машин с набором новых устройств и средств для связи с внешними производственными объектами с приоритетным управлением или с использованием устройства сбора и обработки данных в АСУ и АСУ ТП реализованы в УВК "Днепр-2" (1965—1969 гг.) [5, 6].

Перспективной тенденцией В. М. Глушков считал переход от однопроцессорных ***фоннеймановских*** ЭВМ к многопроцессорным и "умным" интеллектуальным структурам машин [1—4].

Первым шагом на пути создания таких "умных" машин была серия машин для инженерных расчетов "Мир". В ней схемно и программно реализован новый алгебраический язык "Аналитик" для проведения аналитических преобразований, доказательства теорем и решения задач численно-аналитическими методами. На настоящее время на заводе ВУМ (г. Киев) совместно с германскими специалистами изготовлена к 90- летию Глушкова новая интеллектуальная машина, развивающая серию "Мир-3" [10, 11]. Она выполнена под руководством известного специалиста Института кибернетики И. Н. Молчанова. В эту машину встроены программы решения систем линейных, нелинейных, интегральных и дифференциальных уравнений, численного интегрирования задач Коши, краевых задач, кратных интегралов и др. Машина разработана в составе "знание-ориентированного" комплекса "Инпарком". Этот комплекс обеспечивает реализацию новых математических м етодов и организацию параллельных вычислений для решения сложных математических задач с помощью встроенных методов, описанных в статье И. Н. Молчанова из сборника [11].

Язык "Аналитик" и в наше время постоянно развивается в Институте математических машин и систем НАН Украины. В нем активно используют методы компьютерной алгебры, ориентированной на создание математических моделей для исследования технических, компьютерных объектов и процессов обработки информации [4, 12].

Новым по меркам 1970-х гг. видом ЭВМ для управления технологическими процессами на промышленных предприятиях стал УВК "Днепр-2" [6]. Для него было реализовано оригинальное общесистемное программное обеспечение, необходимое для организации разработки систем управления и проведения вычислений задач, возникающих на автоматизированных промышленных предприятиях. В частности:

— ОС с диалоговым, многопультовым режимом разработки, отладки, выполнения программ и управления вычислениями со сбором и обработкой данных в реальном масштабе времени (разработчики Г. Я. Маш-биц, В. И. Конозенко, Е. И. Калайда и др.);

— трансляторы с языков Автокод, АЛГАМС и КОБОЛ (разработчики Е. М. Лаврищева, Л. П. Бабенко, Е. Л. Ющенко и др.), разработанные на основе нового СМ-метода анализа программ [7, 8] с табличным представлением грамматик языков программирования и положивший начало построению синтаксисо-семантических управляемых трансляторов. СМ-метод[1] является усовершенствованием известного м етода синтаксического контроля Замельсона и Бауэра. С помощью СМ-метода реализованы семантика общего арифметического блока транслятора АЛГАМС и КОБОЛ (разработчики Л. Г. Усенко и С. Л. Берестовая), компилятор с подмножества языка SQL в ГДР (1972 г.) и др.

Техническое и математическое обеспечение УВК "Днепр-2" описано в документации на

[1] См. статью E. M. Lavrishcheva and E. L. Yushchenko. A method of analyzing programs based on a machine language, URL: http://link.springer.com/article/10.1007%2FBF01068491

систему (общее описание, руководство программиста, инструкции и др.). Это обеспечение прошло успешные испытания в государственной комиссии (1967 г.) под председательством директора Вычислительного центра АН СССР академика А. А. Дородницына.

УВК "Днепр-2" был представлен на Первой международной выставке ЭВМ в Москве в 1969 г. Огромный интерес к комплексу проявили специалисты в области вычислительной техники из США, Франции, Германии (IBM, CDC, CDS, Bull Jeneral Electric и др.). Им хотелось получить от разработчиков этого комплекса, которые выступали в роли экскурсоводов, новые сведения, понять оригинальные идеи, реализованные в его архитектуре (концепцию прерывания, средств связи и управления объектами предприятий, режим разделения времени и др.).

После этой выставки был подписан контракт по поставке УВК "Днепр-2" (вычислительный комплекс "Днепр-21" и управляющий комплекс "Днепр-22") для построения АСУ ТП управления прокатом металла на металлургических комбинатах (г. Берлин, г. Лейпциг). Во внедрении и реализации АСУ ТП принимали участие разработчики комплекса и программного обеспечения [2, 5, 6]. Виктор Михайлович Глушков был членом межгосударственной программы ГДР и СССР по внедрению УВК "Днепр-2" в вычислительных центрах ГДР и по разработке АСУ ТП для металлургии на его основе. Он практически руководил данным проектом, который был успешно выполнен. Комплекс "Днепр-2" был надежной программно-технической базой АСУ ТП для управления металлургическими комбинатами ГДР до 1991 г.

По результатам создания ОС, системы программирования для УВК "Днепр-2" и системы управления процессами обработки данными в реальном времени работы АСУ ТП основные разработчики от института Кибернетики защитили кандидатские диссертации: Е. М. Лаврищева — "СМ-метод анализа языка Автокод и АЛГАМС"; Л. П. Бабенко — "Транслятор с языка КОБОЛ"; В. И. Конозенко — "Принципы реализации многопультовой и диалоговой отладки программ и решения задач"; Кухарчук А. Г. — "Принципы построения управляющего комплекса "Днепр-21" и вычислительного комплекса "Днепр-22". Научные результаты, полученные при выполнении этого проекта, вошли в сокровищницу мировой компьютерной науки [4, 5, 10, 11].

Технологии автоматизированных систем

Теория построения АСУ была изложена В. М. Глушковым в его монографиях [2, 3], которыми руководствуются многие специалисты и в настоящее время. В 1970-х гг. В. М. Глушков уделял огромное внимание созданию первых АСУ в Украине, Болгарии и ГДР. По его методологии создавались АСУ и информационные системы, к числу которых относятся АСУ ТП Лисичанского химкомбината, Донецкого горно-обогатительного комбината, Львовского телевизионного завода, металлургического комбината ГДР и других предприятий.

Виктор Михайлович Глушков в 1974 г. предложил Кабинету министров СССР проект системы для объединения и централизованного управления всеми автоматизированными системами СССР, в том числе проект объединения АСУ промышленных предприятий страны в общую государственную автоматизированную систему — ОГАС [2, 11]. Тогда этот проект не был поддержан, так как требовал огромных финансовых ресурсов на его реализацию. Следует отметить, что идеологически проект ОГАС, по мнению автора, предвосхитил появление Интернет. Основы теории управления и построения информационных систем (ИС) были изложены В. М. Глушковым в его последней монографии "Безбумажная информатика", продиктованной им в больнице (октябрь 1981 г. — январь 1982 г.) [3]. Многие идеи, принципы и методы В. М. Глушкова были устремлены в будущее. Они развиваются его последователями — учеными и практиками с учетом новых условий, которые предоставляют Интернет и возможности современных общесистемных сред (IBM, MS.Net, Grid и др.). В период глобальной информатизации и компьютеризации методы В. М. Глушкова получили дальнейшее развитие в системе электронного документооборота Академии педагогических наук Украины. По материалам идей, связанных с развитием ИС и систем документооборота в государственных

органах управления, разработан учебник для студентов высших учебных заведений. В нем изложены основополагающие идеи В. М. Глушкова в ИС, новые современные подходы к обработке деловой информации и управлению ИС ч ерез систему Интернет (см. http://Hb.iitta.gov.ua/view/creators).

Технология создания программных продуктов

Появлению современной технологии создания программного продукта предшествовало программирование для решения разных математических задач на первых ЭВМ. Описание программ выполнялось машинным языком, адресным языком Е. Л. Ющенко и языками программирования четвертого поколения, включая Ассемблер, Algol-60, Fortran, PL, Модула, Ада и др. Для их практического использования разрабатывались соответствующие программы для ЭВМ. Идеи и концептуальные положения автоматизации программирования были в центре внимания зав. отделом ИК АН УССР Е. Л. Ющенко. Вместе с аспирантом Г. Е. Цейтлиным им были разработаны теоретические аспекты использования алгебраических методов В. М. Глушкова в программировании, концептуальные положения синтаксического и семантического анализа языков программирования, система алгоритмических алгебр (САА) на основе теории автоматов Глушкова (1972 г.) [10, 12]. В этих работах представлена формальная теория универсальных алгебр, автоматных и алгоритмических САА, контекстно-свободных языков программирования и метаязык СМ-грамматик для реализации семантических программ трансляторов. Первые работы в области теории программирования были выполнены А. П. Ершовым, Г. Е. Цейтлиным и Е. Л. Ющенко [10] и опубликованы повторно, в том числе и на английском языке. Заметим, что Г. Е. Цейтлин постоянно развивал САА, создал теорию алгоритмики, подготовил более десяти кандидатов наук и реализовал систему Мультипро- цессист, основанную на идее многопроцессорности компьютеров Глушкова.

В 1975 г. В. М. Глушков предложил перспективный сборочный способ для постепенного перехода от "ремесленного" производства к промышленному выпуску компьютеров, программ и аппаратно-программных систем. Индустрия компьютерных программ, по его замыслу, должна была базироваться на технологических линиях конвейерного изготовления продуктов. Технологии создания компьютеров, информационных и программных систем он считал движущей силой прогресса фундаментальных кибернетических и компьютерных наук.

В работе [12] В. М. Глушков сформулировал три перспективных направления технологии программирования:

- модульная система автоматизации производства программ по методу сборки (АПРОП) программных заготовок-модулей в сложные системы [7, 8, 13];
- метод формализованных технических заданий для проектирования сложных программных комплексов с использованием семейства алгоритмических языков для описания отдельных блоков систем на уровнях последовательной детализации компьютерного проекта Маяк [2];
- Р-технология программирования для автоматизации конструирования структур программ и данных для программно-технических систем средствами графического Р-языка.

В результате поисковых и прикладных исследований на этих научных направлениях были разработаны новые методы, технологии и инструментальные средства. К их числу относятся: Р-технология и стандарт Р-языка в ISO/IEC (И. В. Вельбицкий [14]); метод сборки разнородных модулей в АПРОП; технология систем и пакетов прикладных программ, отдельные пакеты программ математического, экономического, статистического типов (И. В. Сергиенко, А. С. Стукало, И. Н. Редько и др. [15]). Этими работами был внесен существенный вклад в теорию и практику индустрии создания программных продуктов в СССР на основе метода сборки, сформулированного В. М. Глушковым и реализованного в АПРОП ЕС ЭВМ [7, 8, 13].

Сформировалась технология программирования и новый вид программирования — сборочное программирование для объединения разноязыковых модулей средствами системы АПРОП. Такая система разрабатывалась по договору с Институтом приборостроения (г. Москва) в составе технологии создания программ для бортовых систем "ПРОТВА", которая реализовывалась под руководством В. В. Липаева [16]. Главное нововведение системы АПРОП — интерфейс (межмодульный, межъязыковый и технологический) [8, 13, 17], а также библиотека интерфейсных функций преобразования нерелевантных типов данных, описанных на разных языках и платформах. Впервые автором было определено понятие интерфейса и языка его описания (1976 г.) в проекте АПРОП [7]. Идеи обеспечения связи

разноязычных модулей в АПРОП во многом опередили появление зарубежных языков интерфейсов (M1Ь API, IDL, SIDL и др.). Интерфейс начал использоваться в процессе создания новых программных систем из модулей. Понятие "интерфейс" стало общепризнанным после международной конференции "Интерфейс СЭВ-1987".

Технология сборочного программирования начала формироваться при В. М. Глушкове как средство программирования и сборки сложных прикладных систем, а также семейств трансляторов. Основные положения этой технологии были связаны с выделением общих средств в языках программирования операционной системы единой системы ЭВМ (ОС ЕС), их программной реализацией и сборкой из них трансляторов в системе "Терем" в отделе В. М. Глушкова. В этой системе методом сборки разработанных общих компонентов языковых процессов в классе языков ОС ЕС реализовано семейство языков МАЯК [18].

Становление в СССР сборочного программирования поддерживали академик АН СССР А. П. Ершов [19] и проф. В. В. Липаев в проекте "ПРОТВА" [16]. А. П. Ершов считал, что сборочное программирование эффективно, поскольку готовые программные модули позволяют быстро решить многие задачи из определенной проблемной области.

В дальнейшем сборка стала важным технологическим решением для индустрии создания программных продуктов. Это обстоятельство было подтверждено защитой в 1989 г. автором настоящей статьи доктор - ской диссертации "Методы, средства и инструменты сборочного программирования".

Таким образом, сборочное программирование сложных систем из готовых программных модулей было де-факто сформулировано как новый подход к программированию и, в соответствии с концепцией академика В. М. Глушкова, как основа фабрик для конвейерной разработки программ.

Индустриальная технология сборки программных продуктов

Методика создания технологических линий (ТЛ) предложена учениками В. М. Глушкова, включая автора, в 1987 г. [20]. Она апробирована на шести линиях автоматизированной ИС "Юпитер-470" Института кибернетики АН УССР для военно-морского флота СССР (1983—1991 гг.). Эти ТЛ стали первой практической реализацией идеи сборки ТЛ из процессов и операций, которые отображаются в маршрутной схеме ТЛ.

Фабрики программ разрабатывают как инфраструктуру создания в промышленном режиме программных продуктов с заданными функциями, структурой и качеством. В основе фабрики — большое число разного рода готовых к использованию ресурсов и средства разработки из них продуктов различного уровня сложности и назначения. Основные механизмы фабрики — ТЛ, задающие порядок разработки сложной продукции из готовых ресурсов, которые находятся в хранилище (складе) фабрики или, при необходимости, подбирают из разных библиотек и репозитариев Интернет.

Исходя из опыта автоматизированной сборки разнородных программ в АПРОП и анализа современных зарубежных фабрик индустриального типа [21], автор сформулировала общий набор элементов, которые характеризуют любую фабрику программ:

- готовые ресурсы (артефакты, программы, сервисы, многократные компоненты и т. п.);
- спецификаторы интерфейсов (паспортных данных готовых программных ресурсов), которые описывают в одном из языков интерфейса (IDL, API, SIDL и др.);
- операционная среда, которая содержит программные средства и инструментарий для системной сборки разнородных ресурсов;
- технологические и продуктовые линии производства программной продукции;
- метод проектирования, разработки и сборки готовых ресурсов;
- сборочный конвейер производства программ [21—23].

Следует отметить, что к настоящему времени сложились необходимые условия для решения научных и технических задач в рамках Европейских проектов Grid. В рамках этих проектов функционируют фабрики программ системного и прикладного характера с представленными выше элементами производства программных продуктов [24—27].

Построение ТЛ выполняется на этапе технологической подготовки работ для создания специальной схемы линии (технологического маршрута) с помощью процессов и операций, которые обеспечивают продуцирование элементов системы средствами языков программирования или комплекса соответствующих средств [27, 28]. Линии комплектуют из процессов ТЛ, которые отвечают задачам реализации будущей предметной области, стандартных

инструментальных средств, технологических модулей и комплекса соответствующего нормативно-методического обеспечения. При этом могут дополнительно подбираться готовые прикладные ресурсы, инструментальные средства для реализации отдельных функций или элементов программ.

Согласно положениям SWEBOK процессами являются анализ требований, конструирование, разработка, тестирование, эксплуатация и модернизация. С помощью соответствующих технологических ресурсов их поддержки и видов обеспечения формируется технологический маршрут линии для выполнения отдельных задач разработки и сборки программных продуктов.

Все ресурсы и процессы связаны между собой технологическим маршрутом, который устанавливает порядок операций и процессов, выполняемых разными видами обеспечения (информационным, методическим, математическим и программным). На конечной операции маршрута выполняется операция оценивания качества продукта по принятой методике. Набор процессов формируется с учетом международных стандартов ISO/IEC 12207—96, 2007 и ГОСТ 3918—99. Процессы ТЛ описывают специальным языком со ссылками на соответствующие инструментальные средства, технологические модули и правила управления деятельностью специалистов по выполнению отдельных операций таких процессов.

Построенные ТЛ стали первым вариантом промоделированного сборочного конвейера Глушкова. С помощью ТЛ было создано более 500 программ обработки данных для разных объектов автоматизированной ИС "Юпитер".

Позже (2004 г.) появился альтернативный подход — линии продуктов Института программной инженерии США (http://sei.cmu.edu/productlines/frame_report/). Этот подход основан на интеграции ранее разработанных программных продуктов в семейство продуктов с помощью модели характеристик, общей для членов этого семейства. Подход используется при коммерческой сборке готовых программных продуктов, которые написаны на "старых" традиционных языках программирования. Такие продукты, как правило, сложны для понимания и их трудно модифицировать и эксплуатировать. По этой причине инициаторы этого подхода выдвинули концепцию вариабельности программного продукта (способности продукта к изменениям, адаптация продукта). Согласно ее положениям, к уже созданным продуктам применяют методы реинженерии и реверсной инженерии программных продуктов. Такие методы помогают решать некоторые вопросы, обусловленные сложностью больших программных продуктов, созданных с использованием традиционных технологий. Уменьшению сложности реализации больших программ может способствовать их сборка из готовых стандартизованных программных элементов и объектный подход Г. Буча. В подобной сборке могут быть использованы готовые ресурсы, находящиеся в репозиториях фирм производителей программных продуктов или в сети Интернет. Использование этих подходов значительно упрощает разработку больших систем и снижает сложность их реализации.

Следует отметить, что объектно-ориентированные языки программирования (ООП) и системы программирования в современных операционных средах еще не получили должного распространения в режимах промышленной реализации. Более распространенным подходом при производстве больших систем стал сборочный конвейер — ***continuos integration*** М. Фаулера (2007 г.), который поддерживается в ряде коммерческих проектов компании ЕПАМ. Такие продукты не способны к автоматизированному управлению изменениями, так как создаются не с использованием ООП, а традиционными методами.

Хотелось бы отметить, что вопросу индустрии программных и информационных систем большое внимание уделяет правительство Украины. По его инициативе и при поддержке 17—18 ноября 2011 г. и 25— 27 октября 2012 г. были проведены два международных научных конгресса по инфраструктуре электронного правительства и индустрии программных продуктов. Отмечено, что эта индустрия развивается в основном зарубежными фирмами, которых в Украине более 1000. Они создают продукты силами студентов университетов и институтов. Такие продукты находят потребителей, в том числе и в Украине [28]. На конгрессе поддержана концепция отечественной сборочной технологии и продемонстрирована экспериментальная студенческая фабрика программ Киевского национального университета им. Т. Шевченко (КНУ).

Анализ фабрик программ показал [28, 29], что в мире разработан спектр технологий, претендующих на индустрию программных продуктов. Это мультитехнология К. Чернецкого и К. Айзенекера с лейтмотивом "от ручного труда к конвейерной сборке"; технология И. Бея с автоматизированным взаимодействием разноязычных программ; потоковая сборка — ***use case UML-фабрики*** программ Дж. Гринфильда (США), Г. Ленца

(Германия) и С. Авдошина (Россия); сборочный конвейер М. Фаулера и компании ЕПАМ и др. Общее, что их объединяет — линии (схемы) сборки различных видов программ для массового использования. Для поднятия уровня индустриального производства программных продуктов в отделе "Программная инженерия" Института программных систем НАН Украины (ИПС НАНУ) в рамках фундаментальных проектов и диссертационных исследований были разработаны следующие новые теоретические положения и технологии [21, 22, 30—32]:

- объектного моделирования предметных областей с использованием теории Г. Буча и определения новых функций объектного анализа и алгебры операций;
- компонентного проектирования программных систем с помощью оригинальных м оде лей компонентов, сред и систем, а также внешней и внутренней компонентной алгебры;
- моделей и методов генерации, трансформации и конфигурации готовых компонентов с точками вариантов в интерфейсах в вариабельные и интероперабельные семейства систем для выполнения в гетерогенных средах [31];
- сервисно-компонентного проектирования распределенных прикладных систем с использованием моделей SOA и SCA;
- онтологического представления жизненного цикла (ЖЦ) программной системы стандарта ISO/IEC 12207, общих типов данных стандарта ISO/IEC 11404 для их реализации в целях генерации новых вариантов ЖЦ и данных для конкретных предметных областей;
- тестирования отдельных элементов программ и процессов, а также их оценки на зрелость по модели СММ и качества по стандартным моделям качества.

Основные аспекты технологий представлены линиями (спектром линий) в инструментально-технологическом комплексе (ИТК) ИПС. Каждая из этих линий повышает производительность изготовления как отдельных элементов продукта, так и продукта в целом, улучшает условия работы исполнителей, сокращает число сборщиков и снижает себестоимость выпускаемой продукции [22].

Фабрика программ
к 90-летию академика В. М. Глушкова

Концепция сборочного конвейера Глушкова реализована с участием студентов факультета кибернетики КНУ и МФТИ в виде экспериментальной фабрики программ (URL: http://programsfactory.univ.kiev.ua). Автор осуществляла научное руководство реализацией этого проекта на практических занятиях.

Министерством образования Украины в соответствии с программой Curricula-2004 в 2006 г. было введено обучение студентов вузов Украины по дисциплине "Программная инженерия" наряду с курсом "Технология программирования" [22, 33, 34]. Студенты КНУ и филиала кафедры МФТИ изучают основы этих дисциплин с помощью электронного учебника, представленного студентами на веб-сайте фабрики. Они выполняют лабораторные работы по тематике технологии программирования программной инженерии [24, 26, 35]. Для обучения студентов программной инженерии автором разработаны учебники на русском и украинском языках [20, 33, 34]. Ряд студентов приняли участие в разработке фундаментального проекта ИПС НАН Украины по теории и технологии программных продуктов (2002—2011 гг.). В рамках проекта разработаны новые теоретические положения объектного, компонентного, генерирующего программирования. Усовершенствована практика разработки систем из объектов, компонентов и сервисов на фабриках. Отдельные результаты этой деятельности представлены в журнале "Кибернетика и системный анализ" и переведены на английский язык [9]. По итогам конкурса учебников по программной инженерии, проведенной фирмой Майкрософт-МГУ в Москве в 2006 г., учебник Е. М. Лаврищевой и В. А. Петрухина [33] опубликован при поддержке Министерства обра-

Страна или регион	Посещения	Посещения, %
1. Ukraine	714	61.93%
2. Russia	291 ■ 25,24 %	
3. Kazakhstan	25 I 2,17 %	
4 Belarus	24 | 2,08%	
5. (not set)	22 I 1,91 %	
6. India	и I 095%	
7 United States	9 | 0,78%	
8. Moldova	6 | 052 %	
9. Germany	5 | 0,43 %	
10. Latvia	5 | 0,43%	

просмотреть весь отчет

Google-статистика веб-сайта ИТК (II квартал 2013 г.)

зования России и представлен в Интернете на сайте www.intuit.ru

Фабрика программ демонстрировалась на ряде международных конференций [29, 36, 37]. Она включена в состав комплекса ИТК ИПС веб-сайта http://sestudy.edu-ua.net

Фабрика оборудована линиями, созданными студентами. В их числе технологии программирования на языках C#У8.^E Java; построения программных артефактов и компонентов для информационных и программных систем, их сертификация согласно стандарту W3C и сохранение в репозитории; сборки готовых компонентов и компонентов повторного использования (КПИ) в сложные структуры программных систем; трансформации передаваемых типов данных между КПИ; метрического анализа и оценки качества составных элементов и программных систем, обеспечивающие взаимодействие систем между собой (VS.Net-Java, Java-Corba, VS.Net-Corba); поддерживающие процессы обучения теоретическим и прикладным аспектам программной инженерии по электронному учебнику. С декабря 2011 г. к фабрике обратилось более 10 000 пользователей из разных стран (см. рисунок).

Участвующие в работе по этой тематике студенты выполнили дипломные работы, написали магистерские диссертации и статьи [23, 28, 37, 38].

С участием студентов фабрика ИТК пополнена такими новыми линиями, как генерация прикладных систем в языке DSL (Domain Specific Language); онтология ЖЦ стандарта ISO/IEC 12207, трансформация общих типов данных GDT стандарта ISO/IEC 11404 к фундаментальным типам данных FDT; веб-сервисы для построения распределенных программных систем из готовых ресурсов, включая сервисы.

Результаты развития идей В. М. Глушкова

Технология программирования, инициированная академиком В. М. Глушковым, стала главной стратегической линией исследований и разработок отдела Программная инженерия" (с 1980 г.) в ИПС НАН Украины. Сотрудниками отдела, аспирантами и студентами сформулирован ряд новых научных концепций и методов, используемых в технологии программирования (на основе которых написаны пять кандидатских и одна докторская диссертация), опубликовано более 50 статей. Веб-сайты ИТК ИПС и фабрики программ уникальны в плане представления фундаментальных основ понятия "Программная инженерия" и реализации концепций, моделей, методов и технологий создания программных систем.

Список литературы

1. **Капитонова Ю. В., Летичевский А. А.** Парадигмы и идеи академика В. М. Глушкова. Киев: Наукова думка, 2003. 355 с.
2. **Глушков В. М.** Кибернетика, ВТ, информатика (АСУ). Избр. тр. в 3-х томах. Киев: Наукова думка, 1990. 262 с., 267 с., 281 с.
3. **Глушков В. М.** Основы безбумажной информатики. М.: Наука, 1982. 552 с.
4. **Системы** компьютерной алгебры семейства АНАЛИТИК. Теория, реализация, применение: сб. науч. тр. / под ред. А. А. Морозова, В. П. Клименко, А. Л. Ляхова. Киев: НПП Интерсервис, 2010. 762 с.
5. **Лаврищева Е. М., Никитин А. И., Усенко Л. Г.** и др. АКД — Автокод машины "Днепр-2". Киев: Ин-т кибернетики АН УССР, 1969. 97 с.
6. **Управляющая** вычислительная система "Днепр-2" / под ред. А. Г. Кухарчука и В. М. Египко. Киев: Наукова думка, 1972. 260 с.
7. **Глушков В. М., Лаврищева Е. М., Стогний А. А.** и др. Система автоматизации производства программ (АПРОП). Киев: Ин-т кибернетики АН УССР, 1976. 134 с.
8. **Лаврищева Е. М., Грищенко В. Н**. Связь разноязыковых модулей в ОС ЕС. М.: Финансы и статистика, 1982. 127 с.
9. **Lavrischeva K. M.** Theory and practice of software factories // Cybernetics and Systems Analysis. 2011. Vol. 47, Is. 6. P. 961—972.
10. **Глушков В. М., Цейтлин Г. Е., Ющенко Е. Л.** Алгебра. Языки. Программирование. Киев: Наукова думка, 1974. 318 с.
11. **В. М. Глушков:** прошлое, устремленное в будущее. Сб. трудов. Киев: Академпериодика, 2013. 290 с.
12. **Глушков В. М.** Фундаментальные основы и технология программирования // Программирование. 1980. № 2. С. 3—13.
13. **Лаврищева Е. М., Грищенко В. Н.** Сборочное программирование. Киев: Наукова думка, 1991. 213 с.
14. **Вельбицкий И. В., Ходаковский В. Н., Шолмов Л. И**. Технологический комплекс автоматизации

программ на машинах ЕС ЭВМ и БЭСМ-6. М.: Финансы и статистика, 1980. 253 с.
15. **Редько В. Н., Сергиенко И. В., Стукало А. И.** Пакеты прикладных программ. Киев: Наукова думка, 1992. 317 с.
16. **Липаев В. В., Позин Б. А., Штрик А. А.** Технология сборочного программирования. М.: Машиностроение, 1992. 272 с.
17. **Лаврищева Е. М., Грищенко В. Н**. Сборочное программирование. Основы индустрии программных продуктов. Киев: На- укова думка, 2009. 319 с.
18. **Мищенко Н. М.** О сборочном программировании языковых процессоров // Интеллектуализация программного обеспечения информационно-вычислительных систем. Сб. трудов. Киев: Ин-т кибернетики им. В. М. Глушкова АН УССР, 1990. С. 45—52.
19. **Ершов А. П.** Научные основы доказательного программирования. Научное сообщение в Президиуме АН СССР на звание академика СССР, 1984. 11 с.
20. **Бабенко Л. П., Лаврщева К. М.** Основи програмно! шж- енери. Киев, Знання, 2001. 269 с.
21. **Андон П., Лаврщева К.** Розвиток фабрик програм в шфо- рмацшному свт // Вюник НАНУ. 2010. № 10. С. 15—41.
22. **Лаврщева К. М., Коваль Г. І., Бабенко Л. П.** і др. Нові те- оретичш засади технологи виробництва сшейств ПС у контекста ГП. Електронна монографи. ДНТ1 Украши, ВИНИТИ Росй та ДНТБ, 2012. 277 с.
23. **Аронов А. О., Дзюбенко А. І.** Пщхщ до створення студент- сько! фабрики програм // Проблеми програмування. 2011. № 3. С. 87—93.
24. **Лаврщева К. М.** Компонентне програмування. Теори і практика // Проблеми програмування. 2012. № 1. С. 3—14.
25. **Лаврщева К. М.** Генерувальне програмування ПС і сшейств // Проблеми програмування. 2009. № 1. С. 3—16.
26. **Лаврищева Е. М., Зинькович В. М., Колесник А. Л.** и др. Инструментально-технологический комплекс разработки и обучения приемам производства программных систем, (укр.). Гос. служба интеллектуальной собственности Украины. Свид. о регистрации авторского права. № 45292, от 27.08.2012. 103 с.
27. **Лаврщева К. М.** Онтолотачне подання життевого циклу ПС для загальноі лшй виробництва програмних продукты // X Мь ждународная научно-практическая конференция TAAPSD-2013. Теоретические и прикладш аспекты разработки программных систем. 25 травня — 2 червня, 2013 р. Украгна, Ялта. Кфовоград: ПП "Центр оперативно! полпрафи "Авангард", 2013. С. 81—90.
28. **Андон П. І., Лаврпиева К. М.** Методологи побудови лшш виробництва програмних продукты і гх практичне застосування // Мгжнародний науковий конгресс Пнформацшне сусшльство в УкрашГ. 24—25 жовтня, 2012. Украіна. Юев. Держ агентство з питань науки, шовацш та шформатизаци Украши. URL: http://www.ict-congress.com.ua/attachments/article/102
29. **Kolesnyk A., Clabospitskaya O.** Tested Approach for Variability Management Enhancing in Software Product Line // ICTERI 2012, the 8th International Conference on ICT in Education, Research, and Industrial Applications. 6—10 June 2012. Ukraine, Kherson. URL: http://ceur-ws.org/Vol-848/ICTERI-2012-CEUR-WS-paper-31-p-155-162.pdf
30. **Лаврищева К. М.** Програмна шженери. Киев: Академпе- рюдика, 2009. 371 с.
31. **Андон Ф. И., Коваль Г. И., Коротун Т. М.** и др. Основы инженерии качества программных систем. Киев: Академпериоди- ка, 2007. 680 с.
32. **Лаврищева Е. М.** Основы технологической подготовки разработки и прикладных программ СОД. Препринт 87-5 ИК АН УССРЮ. 1987. 30 с.
33. **Лаврищева Е. М., Петрухин В. А.** Методы и средства инженерии программного обеспечения. М.: МОН РФ, 2007. 415 с.
34. **Лаврищева Е. М.** Методы программирования. Теория, инженерия, практика. Киев: Наукова думка, 2006. 451 с.
35. **Лаврщева К. М., Слабоспицька О. О., Коваль Г. І., Колесник А. О.** Теоретичш аспекти керування варибельшстю в семействах програмних систем // Вюник КГУ, сери фи.-мат. наук. 2011. № 1. С. 151—158.
36. **Lavrischeva E., Ostrovski A., and Radetskyi I.** Approach to ELearning Fundamental Aspects of Software Engineering // ICTERI 2012, the 8th International Conference on ICT in Education, Research, and Industrial Applications. 6—10 June, 2012. Ukraine, Kherson. URL: http://ceur-ws.org/Vol-848/ICTERI-2012-CEUR- WS-paper-17-p-176-187.pdf
37. **Lavrischeva E., Dzubenko A., Aronov A.** Conception of Programs factory for Representation and E-learning Disciplines of Software Engineering // ICTERI 2013, the 9th on ICT in Education, Research, and Industrial Applications. 19—22 June, 2013. Ukraine, Kherson. URL: http://ceur-ws.org/Vol-1000/ICTERI-2013-p-252- 263.pdf