

**ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ
МОДЕЛИРОВАНИЯ И ПРОГРАММИРОВАНИЯ
ТЕХНИЧЕСКИХ, ПРОГРАММНЫХ И
ПРИКЛАДНЫХ ОБЛАСТЕЙ ЗНАНИЙ.
История и пути развития.**

**Международная научная конференция “Численное моделирование в
механике сплошных сред” памяти О.М. Белоцерковского
27- 29 октября 2021**

Доклад

Е.М.Лаврицевой

д.ф.-м.н., почетного профессора МФТИ, гнс. ИСП РАН

История появления Кибернетики и информатики

Н. Винер в своей книге «Кибернетика или управление и связь в животном и машине» 1948 году дал определение новой науки «Кибернетика - это наука об информации, управлении и связях в живой и неживой природе, ДНК структурах интеллекта и др. Компьютер – инструмент этой науки».

Процессы обработки информации и управления ЭВМ основываются на элементах животного мира путем выполнения арифметических операций над данными, отображая мыслительную деятельность человека средствами естественного (символического) языка и помогает исследовать ДНК и разум человека, развивая биологические и генетические области знаний.

В искусственном интеллекте начали исследоваться молекулы ДНК, как совокупность нуклидов, входящих в 46 хромосом клетки человека, которые делятся на атомы и синтезируются. Синтез атомов в макроатомы провел Р. Фейнман (1959) с помощью специального «сборщика». Он считал, что будут созданы миллионы миниатюрных заводиков, на которых "крошечные» станки будут непрерывно штамповать маленькие «детальки" и собирать их в макро механизмы, устройства и новые вещества (нано вещества) с новыми специфическими свойствами. Эти идеи привели в перспективе к созданию ЭВМи устройств для биологических и медицинских систем.

Кибернетика

На первых ЭВМ - МЭСМ (1955), СТРЕЛА(1956), М-20 (1958), УМШН, Урал были реализованы ОС ПО, трансляторы, отладчики, тестировщики программ на ЯП - Адресный и **Algol-60, PL/1, Fortran, Cobol, Ada** и др.

Программы вводились с устройств и ОС ПО управляла их обработкой (трансляция, отладка и др.) с сохранением во внешней памяти (16, 32, 64 разрядной) ЭВМ. Трансформацию описания программ в ЯП в код ЭВМ **делали трансляторы** (ТА1 – ТА4...) с ЯП и обработку - отладчики, сборщики..

Первыми программами для ЭВМ были простые численные задачи с данными для вычисления. Программы запоминались в ЭВМ для решения, в Библиотеках программ, РФАП - склады программ. Для анализ программ в ЯП создан CM метод: [A method of analyzing programs based on a machine language.](#) **E. M. Lavrishcheva** and **E. L. Yushchenko** 1972, **Volume 8, Number 2**, Pages 219-223.

Появились первые книги по программированию на ЭВМ:

- **А.К.Китов, И.А. Криницкий. Электронные цифровые машины и программирование.- Физматиздат.- 1959.-579с. И сборник статей М.-Совеское радио. 1966.-182с.**
- **Е.И Воробьев, А.И.Китов. Введение в медицинскую Кибернетику.-1977.-287с.**
- **Б.В. Гнеденко, В.С.Королюк , Е.Л. Ющенко -Элементы программирование. ГИФМЛ. М.-1961.-348с.**
- **Е.А.Жоголев, И.П.Трифонов. Курс программирования.-М.:Наука.- 1964.- 388с.**
- **Е.Л. Ющенко. Адресное программирование.-ГИТЛ.-1962.-288с.....**

Теория информатики¹⁻⁴

Н.Винер писал- «информатика - природный элемент», «информация, это не теория и энергия».

Ф. Бауэр (1973) определил информатику – как компьютерную науку, которая включает технические связи для определения физиологии, психологии и нейробиологии.

Информатика затрагивает теорию кодирования, теорию информации, экстраполяции и включает:

- цифровые и символьные вычисления, логическое исчисление, теории автоматов, алгоритмов, алгоритмические и интеллектуальные систем.

Эти теории и понятия сформировались в процессе создания первых ЭВМ и способствовали формированию новых научных дисциплин – информатика и компьютерные науки (Computer Science)³⁻⁴.

В период появления первых ЭВМ сформировались идеи интеллектуальных, умных машин для решения математических задач по ее алгебраической постановке в специальном языке Аналитик на ЭВМ Мир 1-3 (1960 годы), Наири (Ереван), СМ1,2 (Минск).

- 1. Норберт Винер. Кибернетика или управление и связь в животном и машине. New York-London 1948; Советское радио. М.: 1961, втор. Изд. 1968. - 325с.**
- 2. Ф. Бауэр, Г. Гооз. Информатика, М.: 1976. - 484 с.**
- 3. Глушков В.М. Кибернетика, ВТ, Информатика. - Избранные труды в 3-х томах. – К.: Наук. думка, 1980, 262.- с, 267 с., 281 с.**
- 4. Е.М. Лаврищева. Кибернетика, информатика и программная инженерия: аспекты развития.- К.: Пробл. Программ, 2010. - с. 1-14.**

Перспективной тенденцией развития вычислительных систем В. М. Глушков считал «переход от однопроцессорных фон-неймановских машин до мозгоподобных по структуре машин». Он один из первых запрограммировал математические методы решения систем дифференциальных и интегральных уравнений, метод Дирихле, Адамса — Штермера и др. (Глушков В.М. Об одном методе автоматизации программирования. — М.: Проблемы кибернетики, 1957, № 2. С. 181–184). Эти программы записывались в Библиотеку ИС-2 на ЭВМ М-20. Перспективным развитием технологии программирования Глушков В. М. считал «технологии комплексного проектирования ВС, при которой проектирование технических средств компьютеров объединено в процесс моделирования ОМО. Эта идея реализована в системе ПРОЕКТ с использованием формализованных технических заданий и в языке Аналитик «Мир-2».

1. Норберт Винер. Кибернетика или управление и связь в животном и машине. New York London 1948; Советское радио. М.: 1961, втор. Изд. 1968. - 325с.
2. Ф. Бауэр, Г. Гооз. Информатика, М.: 1976. - 484 с.
3. Глушков В.М. Введение в теорию самосовершенствующихся систем.- К.: Изд-во КВИРТУ, 1962.-107с.
4. П. Велихов. Информатика – актуальное направление советской науки, 1960,
5. А.А. Дородницын. Информатика. Предмет и задачи, 1961.
6. В.С. Михалевич и др. Информатика – новая область науки и практики, 1961.
7. В.М.Глушков. Основы безбумажной информатики.-1982.

Информатика и прикладная математика

Понятие мягкого, имитационного и симуляционного моделирования для закона сохранения энергии Фурье и решения уравнений в механике сплошных сред ввел О.М. Белоцерковский в своей книге Численное моделирование в механике сплошных сред. –Изд-во Физ.-мат.лит. (1994.-441с.). А.А.Самарский определил понятие модели систем (Компьютер и жизнь. Математическое моделирование. –Москва.-1982.-127с.

О.М. Белоцерковским проведены физические эксперименты методами численного решения задач математической физики, нелинейной механики на ЭВМ. Одной из задач была задача о ядерном взрыве и об обтекании затопленных тел потоком сверхзвукового газа. Была определена разностная схема и проведена проверка ее устойчивости. Применение методов механики сплошных сред привело к решению интегральных и дифференциальных уравнений для природных явлений. Создана приближенная модель, отражающая характерные свойства физического природного явления. Разработка численных моделей проводилась с помощью интеллектуальных гибридных *экспертных систем* для распознавания образов и формального описаний ситуаций. Система *расознавания* образов позволила оценить и описать ситуации явлений.

Интеллектуализация информационных ресурсов средствами экспертных систем начала использоваться в медицине, биологии и в других областях знаний (**Белоцерковский О.М., Холодов А. С.** Медицина в зеркале информатики. М.: Наука. 2008.242с.).

Использование методов в механике сплошных сред в Арктике

Искусственный интеллект сыграл важную роль при решении задач в механике сплошных сред, связанных с освоением запасов нефти и газа в условиях Севера и Арктики с обеспечением безопасности и устойчивости ледостойких платформ, нефтегазопроводов, ледоколов и судов ледового океана, а также при миграции крупных ледовых образований, выполняемых под руководством член-корр. РАН Петрова И.Б. в МФТИ:

Исследование Северных морей и электроразведки углеводородов в шельфовых зонах проводится на современных высокопроизводительных вычислительных системах с использованием среды Cloud Computing, Big Data, средств искусственного интеллекта и системных инструментов: ProCAS; Логос; Wolfarm Mathematics; Anasys 3D; Ansis Maxwell; Fluid Dynamics; Flow vision; Tesseral PRO; Abaqus; Fierdrake; Matlab; Mathematics; MathCAD.

- 1. Петров И.Б. Воздействие льда и воды на оффшорные структуры и прибрежные зоны в Арктике // Научно-технические проблемы освоения Арктики / Под ред. Н.П. Лаверова, В. И. Васильева, А. А. Макоско. М.: Наука. 2015. 490 с.**
- 2. Квасов И. Е., Левянт В. Б., Петров И.Б. Решение прямых задач сейсморазведки в трещиноватых средах методом сеточно- характеристического моделирования. М.: ООО «ЕАГЕ», Геомодель, 2016. 295 с.**
- 3. Lavrischeva E.M., Petrov I.B. Ways of Development of Computer Technologies to Perspective Nano, Agenda - Future Technologies Conference (FTC) 2017, 29-30 November 2017| Vancouver, Canada, p. 539-549.**

БИОЛОГИЧЕСКАЯ ЗАДАЧА - лечение катаракты глаза*

Катаракта — это заболевание, связанное с помутнением хрусталика. Единственным способом его лечения является удаление хрусталика. Различают интракапсулярную и экстракапсулярную методики удаления катаракты. При первой хрусталик удаляют вместе с капсулой, в которой он находится, а при второй удаляют только хрусталик, а капсула остается в глазу.

Среди экстракапсулярных методик к наиболее прогрессивной можно отнести лазерную экстракцию катаракты. Для проведения операции в глаз вводится иголка с расположенным в ней оптоволоконном. При диаметре прокола менее 3 мм можно обойтись без наложения швов. Во время поглощения лазерного излучения выделяется тепло, воздействующее на роговицу. Кроме того, происходит быстрое расширение материала хрусталика, имеющее характер микроразрыва. Это приводит к появлению возмущений в биосреде глаза, распространяющихся к его границам, к сетчатке, что делает ее зоной риска. Изучение этих явлений проводится численным моделированием поведения биосреды глаза во время операции.

** Беклемышева К. А., Васюков А. В., Петров И. Б. Численное моделирование динамических процессов в биомеханике сеточно-характеристическим методом // Ж. Вычисл. матем. и матем. физ. 2015., Т. 55, № 8. С. 96–106.*

Мюррей Дж. Математическая биология. Т. 1. М.: Ижевск. 2001. 774 с.

Информатизация по Ершову А.П.

А.П. Ершов (1972г.) «Информация – это совокупность знаний о фактических данных, зависимостях между ними для познания и воспроизведения человеческого мышления в системе искусственного интеллекта (1972г.).

Информатика, как наука изучает структуру и общие свойства информации, закономерности процессов обмена, обработки, хранения, поиска и распространения научной, финансовой, экономической и другой информации. Информатика дает механизмы реформирования документов на естественных языках и создания информационно-поисковых, информационно-логических, интеллектуальных систем методами системного анализа и искусственного интеллекта.

Информатика рассматривается как комплексная наука, в состав которой входят теория проектирования и функционирования сложных компьютерных, программных и интеллектуальных систем (Базы знаний и Базы данных) и ТП. Термин информатика в СССР начал использоваться в 1960 г. в теории искусственного интеллекта, ориентированного на моделирование умственной деятельности человека¹⁻⁴ в работах ученых:

- 1. П. Велихов. Информатика – актуальное направление советской науки, 1960,**
- 2. А.А. Дородницын. Информатика. Предмет и задачи, 1961.**
- 3. В.С. Михалевич и др. Информатика – новая область науки и практики, 1961.**
- 4. Г.И. Пospelов «Искусственный интеллект – новая информационная технология», 1993, и др.**

Информатика в Computer Science:

Информационные технологии (Information Technology) для управления решениями задач, бизнесом, коммерцией, экономикой, социальной сферой и др.).

Информационные системы (Information Systems) – это системы обработки информации (обмен информацией, документами, данными и др.).

Программная инженерия (Software Engineering -SE) – методы, средства и инструменты разработки, эксплуатации, сопровождения и прекращения использования программ. SE занимает центральное место в пространстве информатики.

Системная инженерия (Engineering System-ES) - это теория, методы и принципы построения ПО компьютеров (ОС, ОМО, трансляторов, интерпретаторов, анализаторов, планировщиков и др.), обслуживания информационных и программных систем, управления данными и др.

Компьютерная инженерия (Computer Engineering -CE) – это теория, методы и средства построения компьютеров, суперкомпьютеров, многопроцессорных и макроконвейерных машин, кластеров и др. Основу CE составляют теории Тьюринга, фон Неймана, автоматов, алгоритмов, математики, логики, анализа и др. и т.п.

1. Russell Shackelford, James H. Cross , Gordon Davies, John Impagliazzo and others. Серия «Учебные программы по информатике», Комиссия Учебных программ по информатике 2005.-70р. ACM, AIS, IEEE-CS, <http://www.acm.org/education/curricula.html>, <http://computer.org/curriculum>.

Инженерия знаний

Знания представляются в Базе знаний (БЗ), содержащей правила вывода и **информацию** о человеческом опыте и знаниях (стандарт ISO/IEC 2382-1:1993)¹.

На основе фактической информации БЗ и правил вывода делается автоматическое умозаключение об имеющихся или вновь вводимых фактах и осмысленной обработки информации².

Современные БЗ работают совместно с системами поиска и извлечения информации, заданной онтологией в виде набора **понятий и отношений**. Онтологии описываются в специальных языках **OWL, FODA, ODM, DSL**. Проведена разработка онтологии домена **ЖЦ ISO/IEC 12207 -2007** в OWL, средствами Protégé 2.3 с выходным результатом в языке XML³.

Подход к автоматизации ЖЦ докладывался на Международной конференции **«Science and Information-2015»** в Лондоне и предложено комитетом IEEE сделать автору патент на идею онтологизации ЖЦ.

Онтологический подход набирает обороты в Семантик Веб Интернет (www.semantic_web.org), как инструмент построения БЗ разных доменов.

1. ISO/IEC 2382-1:1993, Information technology — Vocabulary — Part 1: Fundamental terms.

2. Гаврилова и др. Базы знаний интеллектуальных систем // Учебник для вузов. — СПб.: Питер, 2000.

3. [Lavrishcheva E.M. Ontology of Domains. Ontological Description Software Engineering Domain. The Standard Life Cycle, Journal of Software Engineering and Applications, July 24, 2015.](#)

4. Федоренко Д.Г., Астраханцев Н.А. Автоматическое извлечение новых концептов предметно-специфичных терминов. Труды ИСП РАН, том 25, 2013, стр. 167-178.

Нано понятия в ИИ в эпоху информатизации

Н. Винер сравнивал ЭВМ с **живой клеткой**. В ней молекула ДНК реализуются в долговременной памяти хромосома, состоящей из молекулы и РНК (информационных, транспортных и рибосомальных).

Молекула ДНК – это совокупность нуклидов, входящих в 46 хромосом клетки человека. Главные события клетки ДНК - деление или синтез.

Идею синтеза атомов клетки в макроатомы предложил Р. Фейнман (1959) с помощью **программного сборщика**. Он считал, что будут созданы миллионы миниатюрных заводиков, на которых "крошечные станки будут непрерывно штамповать маленькие детальки" для маленьких приборов, собирая в них макромеханизмы, вещества и устройства. Эта идея привела к современной идеи миниатюризации и получения новых макроэлементов для применения.

Маленькая частица названа **нано или «карлик»** с одним свойством - слипание (синтез) друг с другом, что приводит к образованию новых наночастиц или новых элементов.

Синтез молекул белков ДНК приводит к образованию комплексной наноструктуры с новыми специфическими свойствами¹⁻³. Это свойство используется в медицине, биологии, авиации керамике, металлургии и др.

1. E.M. Lavrischeva, I.B. Petrov, Ways of Development of Computer Technologies to Perspective Nano, Future Technologies Conference (FTC-2017), 29-30 November 2017| Vancouver, Canada.-p.539-547.
2. Лаврищева Е.М. Программная инженерия . Парадигмы, технологии, CASE-средства. 2 ред. Москва- Юрайт, 2017.- 284 с. (bibloo – online.ru).
3. Лаврищева Е.М. Программная инженерия и технология программирования сложных систем. Учебник. 2- издание. Москва – Юрайт, 2018.- 431с. (bibloo – Online.ru).

Нанотехнология

Это **технология**, ориентированная на получение веществ и устройств с заранее заданной атомарной архитектурой (Э.Дреслер).

Атом – это $10^{10} = 1$ нанометра (нм), а бактерии это 10^{-9} нм. Частицы от 1 до 100 нанометров называют наночастицами.

Один из важнейших вопросов нанотехнологии — это группирование молекул, маленьких элементов самоорганизовываться, чтобы получить новый материал или устройство. Это относится к белкам ДНК, которые могут образовывать комплексные структуры синтезом молекул.

В перспективе программные элементы малые по размеру синтезируются к виду «маленьких частиц» с заданной структурой и функциями.

Появились nano элементы - **нейроны, кванторы, графены, кубиты** (МФТИ, МИСиС и др.). **Кубит** состоит из нескольких джозефсоновских контактных элементов, разделенных тонким слоем диэлектрика. Квантовые биты кубита способны выполнять вычисления, которые не доступны современным компьютерам. Созданы маленькие элементы типа **чипа, транзисторов** и др. для создания маленьких приборов компьютерного типа. Из маленьких элементов типа nano, чипов собираются новые технические приборы и устройства, как на сборочном конвейере для применения в дисциплинах e-science (биология, генетика, физика, медицина и др.) и которые будут способствовать улучшению здоровья общества и жизни на земле.

Методы синтеза ДНК

Один из подходов к синтезу молекул ДНК реализован на основе теории распознавания Маркова в диссертации выпускника МФТИ А.В.Островского^{1,2}, представленной на сайте <http://7dragons.ru/bio>.

Метод синтеза молекул ДНК реализуется следующими операциями:

- определения фрагментов генов — экзонов и интронов с заданной последовательностью нуклеотидов гена;
- выявления принадлежности аминокислот белка к пространственным структурам — спиральям, листам и другим образованиям;
- проведение метода синтеза фрагментов ДНК для образования новых структур.

Обе задачи решаются методами, использующими скрытые Марковские модели и их обобщения. Математическая формулировка этих задач, алгоритмы их решения и доказательство математической корректности изложены в диссертации¹.

1. Островский А.В. Методы распознавания ДНК на основе моделей Маркова со скрытыми переменными. – Автореф. диссертации.- ИК НАНУ.-23с.
2. Лаврищева Е.М. Программная инженерия и технология программирования сложных систем. Учебник. 2-издание. Москва – Юрайт, 2018.- 431с. (bibloo – Online.ru).
3. Цыганков В.Д. Вселенский разум и квантовый нейрокомпьютер.- Синтег. Москва- 2002.-171с.
4. Искусственный интеллект в технических системах. Государственный институт физико-технических проблем.- сборник научных трудов под редакцией Лупичева Л.И.-1998.-153 с.

Автоматизация вычислительных программ*

Создаваемые программы для разных областей требовали средств их автоматизации и выполнения. Возникли системы автоматизации (1975): АПРОП, ДИСУП, Альфа, САПР и др.

Автоматизация средства программирования (Альфа), САПР Шура-Буры М.Р., Ершова А.П., Любимского Э.З. Поттосина Н.И. ..

Сборка вычислительных программ начала проводится, как конвейерная сборка на фабрики Форда. Базисом сборки были модули. Связь модулей потребовал определения паспортной информации для описания входных и выходных данных, а также операторов вызова (Call, RMI и др.). **Модули описывались в разных ЯП** и имели паспорт, который определял интерфейс с другими модулями. **Интерфейс** описывался в языке IDL (Interface Definition language) и брокер объектных запросов ORB CORBA реализовывал **интерфейс через посредников stub и skeleton**.

Это описание упрощало процесс сборки модулей, записанных в ЯП (Алгол, Кобол, Фортран, ПЛ-1, Модула, Снобол и др.).

Система АПРОП. Глушков В.М., Стогний А.А., Лаврищева Е.М. и др. 1976.-137с.

- Ершов А.П. Введение в теорию программ.-М.: -Наука.- 1977.-286с.

- Сборник автоматизации программирования Ершова А.П., Шура-

Система АПРОП стала составной частью комплексов ПРОТВА, ЯУЗА, ПРОМЕТЕЙ в ВПК (с 1979), которые переданы в ЕрНУЦ (Ереван, 1985). Основные разработчики проекта ВПК за создание комплексов технических и программных средств **методом сборки** были награждены премией СМ СССР (1985) и ЛЕМ.

Система АПРОП передана по договору о внедрении в 52 организации страны. Она использовалась при разработке проекта ВМФ СССР АИС «Юпитер-470» (1982-1991). В ней автоматизировано 4 объекта ВМФ (флот, флотилия, корабль, подводная лодка) специальным ПО и ВС для обработки информации с радиотехнических, бортовых приборов и устройств с обеспечением безопасности и отражения внешних атак на объектах ВМФ СССР. Автор принимал участие во внедрении созданных средств автоматизации объектов ВМФ в Ленинграде, Одессе, Мурманске Владивостоке в 1989-1990г.

- Ершов А.П. Введение в теорию программ.-М.: -Наука.- 1977.-286с.
- Сборник автоматизации программирования Ершова А.П., Шура-Бура М.Р. , В.П Ильина и др. 1978.

Простым и сложным типам данных современных ЯП

построены алгебраические системы:

$$\Sigma_1 = \{G_{\alpha}^b, G_{\alpha}^c, G_{\alpha}^i, G_{\alpha}^r\},$$

$$\Sigma_2 = \{G_{\alpha}^a, G_{\alpha}^z, G_{\alpha}^u, G_{\alpha}^e\}.$$

Каждый элемент этих классов ТД определяется на множестве их значений и операций над ними:

$$G_{\alpha}^t = \langle X_{\alpha}^t, \Omega_{\alpha}^t \rangle, \text{ где } t = b, c, i, r, a, z, u, e.$$

Операциям преобразования каждого t ТД соответствует изоморфное отображение двух АС с совместимыми ТД двух разных ЯП.

В системах Σ_1 и Σ_2 преобразуются типы данных $t \rightarrow q$ для пары языков I_t и I_q такими отображениями:

- 1) G_{α}^t и G_{β}^q – изоморфны (q задан на множестве, что и t);
- 2) между X_{α}^t и X_{β}^q существует изоморфизм, для которых множества Ω_{α}^t и Ω_{β}^q разные.

Если $\Omega = \Omega_{\alpha}^t \cap \Omega_{\beta}^q$ не пусто, то изоморфизм между $G_{\alpha}^{t'} = \langle X_{\alpha}^t, \Omega \rangle$ и $G_{\beta}^{q'} = \langle X_{\beta}^q, \Omega \rangle$ сводится к случаю 1).

Метод сборки и библиотека интерфейсных функций (64) преобразования ТД в разных ЯП сделаны в ОС IBM, Microsoft, Oberon, Unix... 64 функции использовались при работе с разнородными ресурсами в современных ЯП (C, C+, Ruby, Python, Prolog, Java, Basic и др.) и стали составной частью современного общесистемного ОС ПО IBM, BSD, GNU, MSBuild(.NET), JavaE, ApacheAnt, Grid, Linux, Intel.

Результаты исследований и разработки метода сборочного программирования были защищены в диссертациях:

Лаврищева Е.М. Модели, методы и средства сборочного программирования СОД, 1988 ВАК СССР;

Грищенко В.Н. Методы и средства обеспечения межмодульного интерфейса в классе ЯП, 1990 и др.

Опубликован метод сборки в ряде монографий.

Через 10 лет на основе метода сборки был разработан стандарт ISO/IEC FDIS 24765: Configuration Assembly, 2009, получивший название конфигурационной сборки. Стандарта метода сборки не было сделано в СССР.

Публикации по методу сборки

- 1. АПРОП- Система автоматизации производства программ Глушков В.М., Стогний А.А., Лаврищева Е.М., Борисенко Л.И., Усенко Л.Г. 1976. К.: 137с.**
- 2. Лаврищева Е.М., Грищенко В.Н. Связь разноязыковых модулей ОС ЕС.- М.: Финансы и статистика.- 128с.**
- 3. Лаврищева Е.М. , Грищенко В.Н. Сборочное программирование.-1991.-217с.**
- 4. Липаев В.В., Позин Б.А., Штрик А.А. Технология сборочного программирования.- М.: 1992.-283с.**
- 5. Липаев В.В. Технология проектирования комплексов программ. – М.: Радио и связь, 1992. – 240 с.**
- 6. Лаврищева Е.М. Модели, методы и средства сборочного программирования, 1988.-Автореф. Докор.диссер-37с.и 351с.;**
- 7. Грищенко В.Н. Методы и средства обеспечения межмодульного интерфейса в классе ЯП, 1990.- 21с., 157с.**
- 8. Лаврищева Е.М., Грищенко В.Н. Сборочное программирование. Основы индустрии программных систем..- 2009, -К. 431с.**

Графическое представление систем из функциональных модулей

Структуру системы, приложения предметной области знаний можно задать графом G

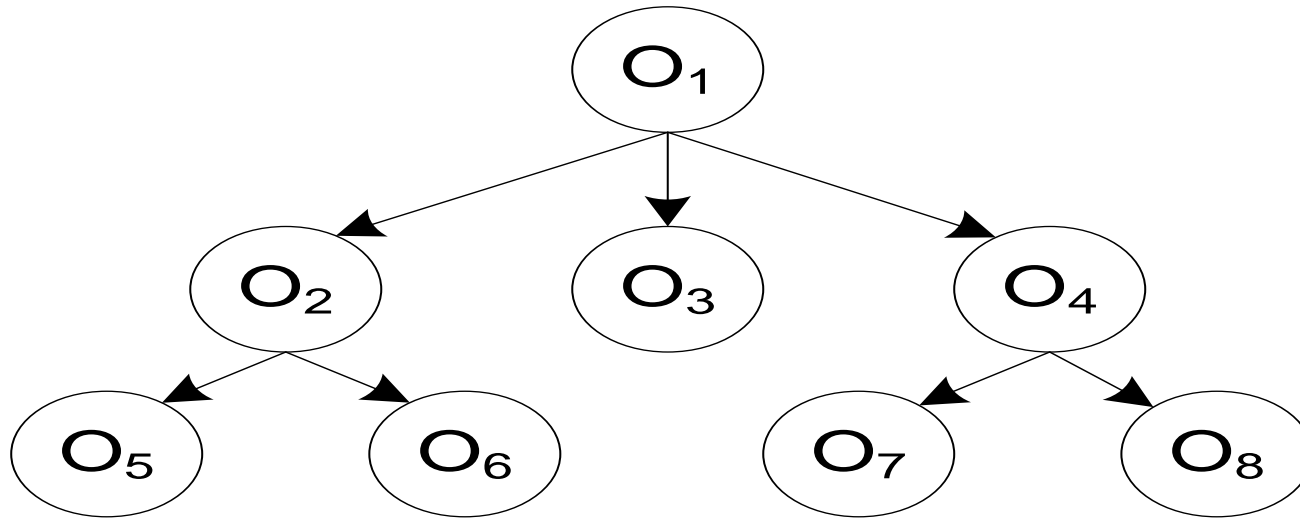


Рис.1. Структура графа $G = \{O\}$

В этом графе: $O_1 - O_8$ – вершины, которые задают взаимно однозначное отображение функций объектов;

- для каждой вершины существует хотя бы одна связь с другой вершиной графа (\rightarrow), задаваемая операцией Link;

- существует лишь одна вершина O_1 графа G , которая имеет статус объекта, отображающего систему в целом.

Аксиома 1. Каждый объект графа имеет хотя бы одну характеристику, которая задает семантику и уникальную идентификацию во множестве объектов системы.

Характеристика содержит свойство, которое определяется типом объекта и принадлежностью его только одному объекту графа. Характеристика может быть **внешней и внутренней**. Любой объект графа может задавать только одну внешнюю вершину для передачи связанному объекту графа и **внутреннюю характеристику**.

Между объектами графа задается отношение **связь-интерфейс**, в котором передаются внешние характеристики объектам, связанным на графе отношением link принадлежности.

К представленному графу G (рис.1) добавляются интерфейсные I объекты: $G = \{O, I, R\}$,

где O – множество объектов-функций, I – множество интерфейсов, R – множество отношений (relations) между объектами (рис.2).

В граф G входят интерфейсные объекты $I = (O'_5, O'_6, O'_7, O'_8)$, которые выполняют вызов Call/RPC/IDL/WSDL объектов и передачу данных другому объекту в требуемом формате платформы ЭВМ. Эти объекты на графе занимают промежуточное состояние O'_5 .

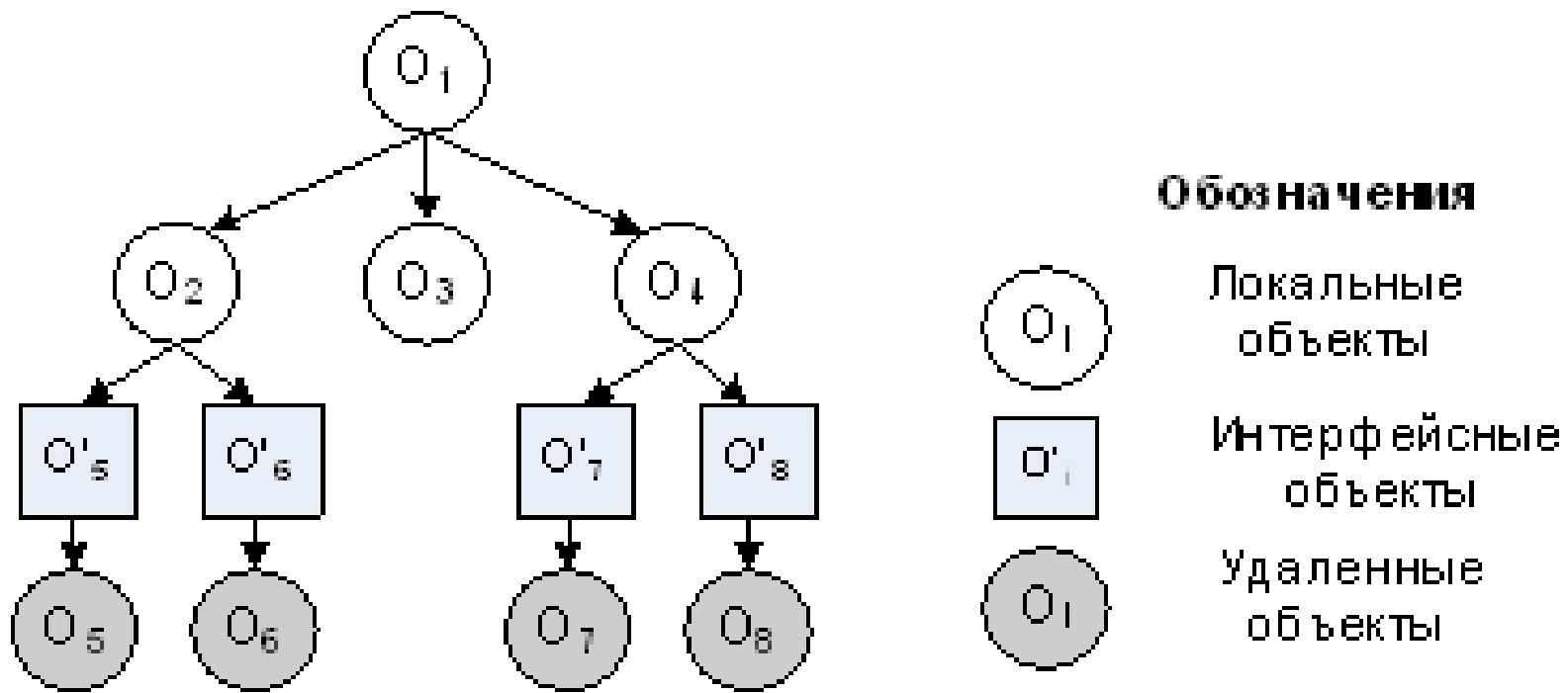


Рис. 2. Граф $G=\{O, I, R\}$

Элементы графа $O_1 - O_8$ описываются в ЯП, а интерфейсы $O'_{25}, O'_{26}, O'_{47}, O'_{48}$ в специальном языке интерфейса **IDL** (Interface Definition Language), WSDL, API, OWL.

Параметры внешних характеристик передаются между объектами через интерфейсы и помечаются как *in* (входной), *out* (выходной), *inout* (входной и выходной).

Интерфейсы в графе задают описание передаваемых данных, задаваемых в операторах удаленных вызовов RPC/RMI/Call с передачей данных во внешних параметрах.

На основе графа с помощью математических операций (объединения, соединения, разности и др.) проводится сборка **link** сложных программных структур (комплекс, агрегат, система и др.) и доказательство их достижимости. Графовая теория реализована в ВПК под руководством Липаева В.В. (1975-1985) и при поддержке академика А.П. Ершова в ИПИ СО АН СССР, ученики которого развивали теорию графов для программирования трансляторов и операционных систем.

Начиная с 2003 года, теория графов начала применяться для моделирования сложных систем из объектов и компонентов с применением новых операций сборки (**assembler, config, make, build**) согласно стандарта **ISO/IEC 828 Configuration (1996, 2012)** и с применением аппарата изоморфного отображения неэквивалентных типов данных (структурных, неструктурных и генерируемых), передаваемых между отдельными объектами создаваемых программных структур по графу. Метод графового моделирования систем из информационных и интеллектуальных ресурсов используется для создания современных вариантов ядра ОС Linux (**Международная конференция ОС DAY-2021**) для систем реального времени, а также в таких прикладных областях знаний, как медицина, генетика, биология и др.

Пути развития искусственного интеллекта в Интернет¹⁻³

1. Web 4.0 будет обеспечивать взаимодействие участников сети с помощью **интеллектуальных агентов**.
2. Корпоративные решения для **облачных технологий** (PaaS, SaaS), будут сращиваться с Интернет-пространством и использоваться для управления адаптивными Web-приложениями, в которых облачные сервисы будут взаимодействовать через **веб-страницы агентов**.
3. **Internet of Things, Smart IoT** будут поддерживать конкурентные APPS через: распределенные microservices; Hypercat мобильной связи; GSM-R для цифровых дорог. Промышленный Интернет (Industrial Internet) ориентируется автоматизацию новых концептов - **«умная энергетика», «умный транспорт», «умные приборы», «умная промышленность», «умные дома», «умные города и медицина»** и т.п.
4. Интернет 4.2 предоставляет технологическое **конфигурирование и взаимодействие** с устройствами обработки больших данных для превращения **Big Data в Smart Data**.
5. Семантика взаимодействия информационных объектов, которые задают **цифровые проекции** реальных или абстрактных объектов, будут взаимодействовать через **Web-сервисы** в языке OWL стандарта ISO 15926 Интернета 3.0.

¹ Е.М. Lavrischeva, I.B. Petrov, Ways of Development of Computer Technologies to Perspective Nano, Future Technologies Conference (FTC-2017), 29-30 November 2017| Vancouver, Canada.-p.539-547.

² Боровков А.И. Компьютерный инжинеринг Учебное пособие // Боровков А.И., Клямин О.А., Мельникова М.П. и др. /Computing Engineering.- San-Peterburg.-2012.- Россия, Полит. ул. 29.- 93 с.

³ Пройдаков Э.А. и др. Платформа «АН2», 2016, 139с.

Теория технологии программирования (ТП)

Теорию ТП как математическую дисциплину сформулировал А.П. Ершов и зарубежные ученые (Е.Дijkstra С. Hoar, Z. Manna, D.Gries, D.Vuirner, К. Parnas и др.). Эта теория основана на теории множества, алгебре, логике исчисления предикатов, комбинаторике и др. Формальная нотация программ ТП – это задание семантики в виде совокупности нетривиальных фактов о вычисляемых ею функциях. В докладе на звание академика АН СССР (1986) и на Всесоюзной конференции «Технология программирования» (1987) А.П.Ершов определил основные методы программирования ТП:

синтеза, сборки и конкретизации.

Синтезирующее программирование базируется на методе доказательного рассуждения о правильности программы.

Сборочное программирование осуществляет построение программы из уже существующих (проверенных на правильность) готовых фрагментов программ (КПИ, reuses) и сборку их в сложную структуру.

Конкретизирующее программирование обеспечивает построение системы по универсальной модели для некоторой предметной области.

1. Ершов. А.П. Научные основы доказательного программирования.– Доклад АН СССР, 1985.–с.1–14.

2 Ершов А.П. Отношение методологии и технологии программирования, Конф. «Технология программирования» и ж. Программирование , 1986, № 3.

Перспектива развития ТП

Теория ТП по Ершову - «это конкретный способ организации, создания, распространения и сопровождения программного продукта (ПП) и деятельность профессионалов, разрабатывающих ПП.

Технология и методология – это *наука*, в то время как метод входит в них составной частью (отчуждаемость и тиражирование ПП). **ТП** – это совокупность методологических положений, организационно-административных и инструментально-технических средств, их информационного и ПО, регламентирующего деятельность людей в разработке ПП». Ершов А.П. определил для ТП (1986) три направления развития на перспективу:

- **1 направление** (*организационное программирование*) 1975–1985 гг.
- **2 направление** (*сборочное программирование*) 1985–1995 гг.
- **3 направление** (*доказательное программирование*) 1995–2005 гг.

В заключительной части он сказал: «Было бы полезно выработать норматив по ТП, который, не затрагивая конкретного методологического или языкового наполнения, унифицировал бы общую этапность разработки ПП; нормативы производительности и надежности; документацию и среду вычисления; **межмодульный интерфейс** поддержки **сборочного программирования...».**

Синтезирующее программирование¹⁻³

Синтезирующее программирование — это процесс получения программы из условия задачи и метода ее решения. Синтез программ основан на логической и аналитической спецификациях.

Спецификация трактуется как формулировка теоремы, утверждающая существование решения задачи.

Синтез программы состоит в доказательстве теоремы существования и преобразования спецификации в машинную программу решения задачи.

При аналитическом подходе спецификация трактуется как уравнение символьного преобразования операций программы с проверкой правильности.

Синтез программы — это некоторый способ манипулирования знаниями, воплощенными в условии задачи предметной области и универсальным знанием, отражающим общематематические закономерности и сущность доказательного рассуждения.

1. Кахро М. И., Калья А. П., Тыгу Э. Х. Инструментальная система программирования ЕС ЭВМ (ПРИЗ). М.: Финансы и статистика, 1981.

2. Тыгу Э. Х. Концептуальное проектирование. М.: 1984.- 287 с.

3. Лавров С.С. Синтез программ. Кибернетика, М.: 1982.-№6.-с.11-16.

4. Лавров С.с. Лекции по теории программирования.- Санкт-Петербург.-1999.-107с.

Композиционное программирование

Композиция программ - это операции объединения функций и данных вида: «данные–функция–имя» и «функция-композиция-дескрипция» на множестве именованных данных, дескрипций и денотатов (значений). Операции композиции образуют подкласс стандартных композиций и композиционных функций на уровне ЯП.

Теория дескриптивных и декларативных программных формализмов обеспечивает адекватное моделирование структур данных, программ и средств их конструирования. В основе этой теории лежит семантика, система композиций и номинативных данных для описания функций и композиций.

Теория тесно связана с алгебрами функций и данных и отличается от традиционных систем программ теоретико-функциональным подходом, классами однозначных n -арных функций и номинативными структурами данных типа множества, мультимножества, реляции и т.п.

1. Редько В.Н. Композиции программ и композиционное программирование // Программирование. – 1978. – № 5. с. 17–26).
2. Никитченко Н.С. Номинативно-эмпликативное программирование. Проб. Програмування. –1999.– № 1, с. 16–31.
3. Никитченко Н.С., Шкильняк С.С. Математическая логика и теория алгоритмов. ВПШ КНУ, 2008.-528с. ISBN 966-439-007-0.
4. Лаврищева Е.М., Никитченко Н.С., Омельчук Л.П. Технология программирования информационных систем. ВПШ КНУ.- 2017.-431с.

Функциональное программирование – РЕФАЛ¹⁻²

РЕФАЛ применяется для решения задач, связанных с распознаванием образов, общением на естественном языке, реализацией экспертных систем, доказательством теорем, символьными вычислениями и др. (теория лямбда-исчисления А. Черча и комбинаторная логика Х. Карри).

Функциональная программа включает совокупность *определений функций*, вызовов других функций и управляющих вызовов. **Рекурсия** – это фундаментальная основа построения семантики. В языке РЕФАЛ (**В.Турчин**, 1966) определил функцию, как упорядоченный набор *предложений*, состоящих из *образца* и *шаблона*. На вход функции подаётся *выражение*, которое сопоставляется с образцом и при успешном сопоставлении, это выражение будет результатом функции либо фиксируется ошибка.

Выражение – это последовательность термов, букв, цифр и символ-меток, макроцифры — символьная (цифровая) запись, неотрицательных целых чисел, чисел с плавающей точкой, *переменных* из одного и нескольких *предопределённых типов*, с произвольным выражением в круглых скобках или в «угловых» скобках с вызовом функций. Этот язык развивается до сих пор и используется практически².

1. Турчин В. Ф. Алгоритмический язык рекурсивных функций (РЕФАЛ). — М.: [ИПМ АН СССР](#), 1968.
2. Смирнов В.К. [Аппаратная реализация языка Рефал в ИПМ им.М.В.Келдыша](#) //Препринт ИПМ им. М.В.Келдыша. — 2003. — № 99. — С. 1-21.
3. Гурин Р.Ф., Романенко С.А. Язык программирования Рефал Плюс. — Переславль: Университет, 2006. — 13 с. — [ISBN 5-901795-08-3](#).

Расслоенное (аспектное) программирование¹⁻²

Технология рассредоточенных действий или технология вертикального слоения разработана **А. Л. Фуксманом**¹. По этой технологии вертикальный слой (срез) содержит совокупность рассредоточенных действий, фрагментов кода, которые реализуют определенную *расширяемую функцию*, а процесс разработки и модификации программы – это последовательность операций добавления или изменения этих функций.

Эта концепция положена в основу новой стратегии поэтапной разработки программ. В ней на первом этапе создается "основа" - упрощенная версия программы. На последующих этапах реализуется программа и она добавляется к вертикальному слою.

В программе могут содержаться "заглушки" – имитаторы отсутствующих частей для вставки готовых программ. Транзакция оформляется как добавление нового компонента (вертикального слоя), который распадается на модули, предназначенные для нескольких горизонтальных слоев. Любая программа имеет серии расширений на горизонтальном слое. Данное программирование получила развитие в аспектно-ориентированном программировании за рубежом².

1. Фуксман А.Л. Технологические аспекты создания программных систем. М.: Статистика, 1979. 183 с.

2. Аспектно-ориентированное программирование. - AspectJ (<http://aspect.org>).

3. Gregor Kiszales, John Lamping, John Irvin, Europe Conference on OOP (ECOOP-1997) Finland, Springer Verlag INCS 1241, June 1997.

Агентное программирование (АП)¹⁻⁵

Парадигма АП основывается на теории переписывания термов с помощью системы равенств алгоритма вычислений. Результат вычислений - терм, полученный с помощью *графовых термов* представления данных и знаний о ПрО¹⁻³.

АП позволяет определить поведение систем и их эквивалентность в транзитивной системе. Поведение задается выражениями алгебры $F(A)$ на множестве операций алгебры A . Две операции префиксинга $a \cdot u$ задают поведение u на операции a и недетерминированном выборе $u + v$ одного из двух поведений u и v , которое является ассоциативным и коммутативным.

Конечное поведение задается константами: Δ , \perp , 0 , обозначающими успешное завершение, неопределенное и тупиковое состояние. Алгебра поведения частично задается отношением \leq , для которого элемент \perp – наименьший, а операции алгебры поведения – монотонные. *Среда E* определяется как агент в алгебре действий A и функции погружения от двух аргументов $Ins(e, u) = e[u]$. Первый аргумент – это поведение среды, второй – поведение агента, который погружается в эту среду в заданном состоянии. Алгебра агента – это параметры среды, а значение функции погружения – это новое состояние этой среды.

В АП используется процедурное, функциональное и логическое программирование. Граф термов задается элементами многоосновной алгебры данных (групп, колец, др.). АП было реализовано на ЭВМ Мир 1-3, а агентная теория используется при принятии управленческих решений³⁻⁴.

1. Капитонова Ю.В., Летичевский А.А. Методы и средства алгебраического программирования // Кибернетика. – 1993.– № 3. – С. 7–12.

2. Глушков В.М., Цейтлин Г.Е., Ющенко Е.Л. Алгебра. Языки. Программирование.- К.: 1974.-321 с.

3. Letichevsky A.A., Gilbert D.R. A General Theory of Action Language // Кибернетика и системный анализ. – 1998. – № 1. – С. 16–36.

4. Транхтенгерц Э.А. Взаимодействие агентов в многоагентных средах.- Автоматика и телемеханика.- М.: 1998.- № 8.- с.3-52.

5. Трахтенгерц Э.А. Компьютерные методы реализации экономических и информационных управленческих Решений. Том 1. Методы и средства. Том 2 - Реализация решений. - М.:2009, ISBN 978-5-89638-113-6.

Графовое программирование¹⁻⁶

Теория графов применительно к программированию начало развиваться в школе А.П. Ершова (Котов В.Е., Касьянов В.И, Иткин В.Э., Евстигнеев А.А. и др.). Были выпущены работы по теории схем программ¹⁻⁵ и по применению теории графов при представлении модульных структур программ матрицами достижимости, которые эквивалентны матрицам смежности ориентированного графа в сборочном программировании⁵⁻⁶. Графовая теория активно развивается в ИСП РАН для схемной организации памяти в OS Linux и используется в авиации и медицине, биологии. Эта теория имеет перспективу развития.

1. Ершов А. П., Введение в теоретическое программирование М., 1977.
2. Евстигнеев А.А. Применение теории графов в программирование. Москва.-Наука.- Редакция физ.- мат. наук, Под редакцией А.П. Ершова. 1985 г. 351с.
3. Котов В. Е., Введение в теорию схем программ, Новосибирск, 1978.
4. Кристофидес Н. Теория графов. Алгоритмический подход.-М.: Мир. 1978.
5. Лаврищева Е.М., Грищенко В.Н. Сборочное программирование, Основы индустрии программных продуктов,- 2009.-Наук.думка.- 471с.
6. Лаврищева Е.М. Теория объектно-компонентного моделирования изменяемых программных систем.-www/ispras.ru/ . ISP RAN, 2016, ISBN 928-5-91474-025.
7. И.Б. Бурдонов, А.С. Косачев, В.В. Кулямин Теория соответствия для систем с блокировками и разрушениями. Москва, Физматлит, 2008.-411с.
8. И.Б. Бурдонов, А.С. Косачев, Сортин А. Распределенные алгоритмы на корневых неориентированных графах. Труды ИСП РАН, 2007. Том.29 выпуск.5. Стр.283-310.

Технология программирования и конференции ТП в СССР¹⁻³

Определение технологии программирования было дано в статье Глушков В.М. «Фундаментальные исследования и технология программирования». Ж. Программирование. М.: 1980г. №2.-с.3-13.

Вельбицкий И.В. Технология программирования. К.: Техника, 1984. 277 с. После смерти В.М. Глушкова (1982) Вельбицкий И.В. был ответственным за проведение Всесоюзных конференций ВКП I-III (1982, 1986, 1992)¹ , ИК АН УССР.

Эти конференции собирали тысячи специалистов страны. На них выступали с научными докладами по технологии программирования: Ершов А.П., Липаев В.В., Лавров С.С. Шура-Бура М.Р. И многие другие.

-
- 1. Глушков В.М., Цейтлин Г.Е., Ющенко Е.Л. Алгебра. Языки. Программирование.-К.: 1974, 321 с.**
 - 2. Материалы конференций находятся в музее ВТ (<http://www.computer-museum.ru>).**
 - 3. ВКП I-II. Технология программирования.-1982 (1986).- М.; ГКНТ, ГКВТИ. Труды ВПК I - II.-381с., 457с.**
 - 4. ВКП III. Технология программирования.-1992.- Труды ВПК III.- 587с.**

Развитие ТП в период 1992-2021

Отечественные ТП прошли долгий путь развития и сыграли важную роль в истории программирования программных и системных комплексов различного назначения на ЭВМ (особенно в ВПК). Созданные и используемые технологии не были стандартизированы у нас в стране.

После 1992 года стали использоваться в основном зарубежные технологии - UML (Unified Modeling Language), OWL (Ontology Web Language), WSDL (Web Service Definition Language), XML, XSL, RDF, Semantic Web и др.

Новые ЯП – C++, C, C#, Basic, JAVA, Ada, Prolog, Python и др.

С 2004 года сформировались модели (**Model Variability**), on ProductLine/ProductFamily производства программ, в Grid-системах и применяется на фабриках программ AppFab Общесистемных сред Интернет.

Особое развитие получил научный сервис в среде Семантик Веб Интернета (www.semanticweb.org), позволяющий производить композитную сборку сервисов (SOA, SCA) и Reuses в Веб-системах.

Разработаны технологии системного программирования, интеллектуальные средства (Data Mining) для анализа данных, извлечения знаний (Mining knowledge) и данных в разных прикладных системах, **Cloud Computing, Big Data**, и др. размещаемых в новых глобальных хранилищах.

Проект информатизации России 1992

В 1992 г. создан Научно-технический Совет «**Информатизация России**» для создания новых фундаментальных основ информатизации. В совет входили 30 ведущих специалистов от РАН (Иванников В.П., Липаев В.В. и др.) и от других НИИ России. Совет разработал основы информатизации России:

1. Экономические и социальные задачи информатизации.
2. Техничко-экономическое развитие рыночной экономики информатизации.
3. Научно-техническое обеспечение информатизации России (Фундаментальные разработки ИТ и систем); технология и программная инженерия разработки систем; Обеспечение защиты информации; Стандартизация средств создания систем; поддержка информатизации в образовании; развитие инфраструктуры информатизации России.
4. Развитие инфраструктуры информатизации России.
5. Системный проект информатизации России.

В 3.1 к научным направлениям информатизации отнесены (Иванников В.П.):

- алгебраические и логические теории формализации систем;
- анализ параллельных и распределенных систем;
- формальные спецификации и верификации сложных систем;
- парадигмы программирования;
- формализмы представления знаний;
- модели человеко-машинных интерфейсов и визуализации.

Методология и технология программирования (Проект 1992)

Программная инженерия. Разработка больших программных проектов требует повышенной надежности программ и качества.

Применяемые методологии и технологии CASE обеспечат **эффективную** поддержку всех этапов ЖЦ больших ПС, начиная от эскизного проектирования, математического моделирования до реализации и сопровождения программ.

Результаты такого подхода можно видеть по качеству программных продуктов, поступающих с Запада в нашу страну.

Параллельные вычисления. Создание оптимизированных компиляторов с языков высокого уровня, новых алгоритмических языков, распределенных операционных систем для векторных супер-ЭВМ и ЭВМ с распределенной параллельной архитектурой.

Машинная графика. Создание гипер- и мультисреды, искусственная реальность и графических суперкомпьютеров и рабочих станций на основе параллельных архитектур. Визуализация – это инструмент исследований в физике, биологии, химии, медицине, архитектуре, искусстве.

Действующие парадигмы программирования

К современным *парадигмам программирования* относятся:

- объектная,
- компонентная,
- сервисная,
- аспектная,
- функциональная,
- агентная и др.

За рубежом разработаны технологии:

- Agile, EX, SCRUM и др.

Сущность каждой парадигмы – формальное описание программных и технических элементов для повторно используемых компонентов (КПИ) или Reuses типа – модуль, объект, компонент, сервис и др.

1. Городняя Л.В. Парадигмы программирования: анализ и сравнение.-СОРАН, 2017.-239 с.
2. Лаврищева Е.М. Программная инженерия. Теория программирования. –2016.- МФТИ. - Методичка -51 с.
3. Лаврищева Е.М. Программная инженерия. Технология программирования.-2016 – МФТИ.- Методичка - 52 с.
4. Лаврищева Е.М. Программная инженерия. Технология программирования.-2016 – МФТИ. - Методичка.52 с.

Парадигма объектно-ориентированного программирования

ООП — парадигма основана на представлении предметной области в виде системы взаимосвязанных абстрактных объектов и их реализаций. **Принципы ООП:**

- классы (классификация) объектов, суперклассы;
- *наследование* — механизм установления отношений «потомок — предок» (порождать один класс от другого с сохранением всех свойств и методов класса-предка);
- *инкапсуляция* — свойство сокрытия реализации класса;
- *абстракция* — описание взаимодействия в терминах сообщений/событий предметной области;
- *полиморфизм* — возможность подмены в сценарии взаимодействия объектов одного класса другим объектом со сходной структурой.

Буч Г. Объектно-ориентированный анализ. М. : Бинум, 1998;

Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя : пер. с англ. М. : ДМК, 2000.

Парадигма событийная

Парадигмы событийно-управляемого программирования основываются на концепции организации «внутреннего» потока управления путем выполнения инструкций программы, не связанных с наступлением событий вне ее.

Событийное программирование (*event-oriented, event-based* или *event-driven programming*), которое заключается в подмене системных функций-триггеров «по умолчанию» нужными функциями, соответствующими каждому возможному событию. Программа состоит из ряда маленьких прикладных программ, называемых *обработчиками событий* и *диспетчера* обработки очередей событий и имитирующих среду, управляемую прерываниями. Тестирование операционных (сценарных) профилей программ и прогноз размера ПП по FPA.

Согласованное программирование и параллельные вычисления

Для повышения производительности компьютеров проводится распараллеливание потока команд или потока данных.

Распараллеливание данных подразумевает применение одной операции сразу к нескольким элементам массива данных, а параллелизм задач — разбиение вычислительного процесса на несколько подзадач (процессов), каждый из которых выполняется на своем процессоре.

Согласованное программирование (*concurrent programming*) — это взаимодействие некоторых одновременно работающих вычислителей (нескольких процессоров или многоядерных процессоров). Подходы - *обмен сообщениями*, использование *разделяемой памяти* и *стандарт POSIX*.

Параллельное программирование (*parallel programming*) или параллельные вычисления — *parallel computing*) включает параллельные потоки, порождение и обработка асинхронных событий.

Рыбаков А., Золотарев С. Программное обеспечение многоядерных систем. URL: <http://old.osp.ru/os/2006/02/012.htm>.

Программирование по прототипу

В этом стиле программирования понятие класса отсутствует, а повторное использование (наследование) производится путем **клонирования** существующего экземпляра объекта, как прототипа.

Стиль прототипного программирования – обеспечение поведения небольшого количества «образцов», как «базовых» объектов при создании других объектов.

Создание нового объекта выполняется путем *клонирования* существующего объекта либо создания объекта «с нуля» и дальнейшего получения его копии — клона, который наследует все характеристики своего прототипа и становится самостоятельным, может изменяться без побочных эффектов.

Каноническим примером прототипного языка является язык [Self](#).

Прототипный стиль программирования положен в основу таких языков программирования, как [JavaScript](#), [Squeak](#), [NewtonScript](#), [MOO](#), [REBOL](#), [Kevo](#) и др.

Agile-технологии

Agile Software Development. Включает методологии eXtreme Programming - XP (экстремальное программирование), SCRUM, Dynamic Systems Development Method (DSDM), Adaptive Software Development, Crystal, Feature-Driven Development, Pragmatic Programming.

Методологии Agile ориентированы на тесное взаимодействие команды разработчиков с пользователями на основе итеративной модели ЖЦ с приращениями и быстрой реакции на изменяющиеся требования и ProductLine подход к разработке ПО. В **XP** любой член группы может изменить не только свой код, но и код другого программиста и каждый модуль снабжается автономным тестом (*unit test*), обеспечивая возможность регрессионного тестирования модулей. Тесты пишут сами программисты и они имеют право написать теста для любого модуля.

SCRUM — гибкая методика управления проектом с процессом разработки - анализ требований, проектирование, программирование, тестирование (<http://agile.csc.ncsu.edu>).

DSDM - метод разработки информационных систем консорциума DSDM.

Agile — манифест разработки ПО. www.agilemanifesto.org/iso.

Технология и инженерия программ

Технология программирования – это совокупность методов, способов, средств автоматизации и порядка их использования при сборочном производстве материальных и нематериальных продуктов из элементарных программных элементов (как «материалов» в компьютерах) нематериальной природы (В.М.Глушков, 1980).

Инженерия программ (Software Engineering) – это система методов, способов и дисциплин планирования, разработки, эксплуатации и сопровождения программного обеспечения (ПО), предназначенных для промышленного производства www.swebok.com, 2001, www.semat.com, 2009, 2012.

Разработаны технологии производства трансляторов, языковых процессов, парадигм программирования, фабрики программ (AppFab в VS.Net, IBM,Java, Cobra, Intel,Grid и др).

Фабрики программ

- **Фабрика программ - это инфраструктура ТЛ и ресурсов (технических, программных, человеческих, организационных) для изготовления приложений из ресурсов, КПИ, Reuses путем:**
 - анализа и моделирования целевого домена, ПрО, АСУ;
 - накопления артефактов и КПИ в репозиториях и библиотеках;
 - сертификации КПИ, модулей в ЯП и интерфейсов (IDL, API, WSDL);
 - изготовления элементарных КПИ в ЯП на простых ТЛ;
 - сборки КПИ в сложные ПС в средах (VS.Net, IBM);
 - тестирования КПИ и ПС;
 - передачи ПП на сопровождение и эксплуатацию заказчику.
- Фабрики сформировались в мировой практике:
- фабрика "ручной" сборки разноязычных программ И.Бейя;
 - фабрики потоковой сборки по UML Дж. Гринфильда;
 - фабрика Г. Ленца на основе Use Case;
 - каркас фабрики программ Авдошина;
 - фабрика «continiuos Integration» Фаулера;
 - фабрика сборки программ ЕПАМ [29] и др.

Фабрики программ основаны на:

- Готовых ресурсах (артефакты, КПИ, компоненты, сервисы и др.);
- Спецификаторах КПИ и интерфейсного посредника (паспортных данных, операций вызова с передачей данных);
- метод разработки, композиции, сборки КПИ;
- технологических и продуктовые линии производства ПП;
- компоузере КПИ и других разнородных ресурсах (AppFab);
- сборочном конвейере производства программ;
- операционная общесистемная среда для поддержки сбора и интеграции элементов.

Построен инструментально-технологический комплекс (ИТК) (<http://sestudy.edu-ua.net>), <https://7dragons.ru/ru> и экспериментальная фабрика программ КНУ (<http://programsfactory.univ.kiev.ua>).

Исходя из статистики Google ими пользуются с 2012г. более 100000 с разных стран (США, Россия, Бразилия, Португалия, Белоруссия и др.).

Фабрика сборки разнородных программ¹⁻⁴

К 1975г. модуль стал главным элементарным, отдельным элементом, выполняющим некоторую отдельную функцию, подфункцию в общей структуре большой задачи. Описание программ (модулей) в ЯП переводилось в код ЭВМ отечественными трансляторами, который выполнялся в среде ОС ЕС ЭВМ. Модули были составной частью разработанной структуры создаваемой системы и возникла необходимость их объединять, собирать в структуру конфигурации.

Академик В.М.Глушков высказал идею (1975) о том, что программы будут собираться **конвейерным способом**, как автомобили из готовых деталей на фабрике Форда. С его участием был разработан метод сборки модулей с помощью **межмодульного интерфейса** (1976–1982) **в системе АПРОП¹**.

Межъязыковый и технологический интерфейс получил грамоту на конференции «Интерфейс СЭВ (1987) и он стал базовым понятием в технологии **сборочного** программирования³⁻⁴.

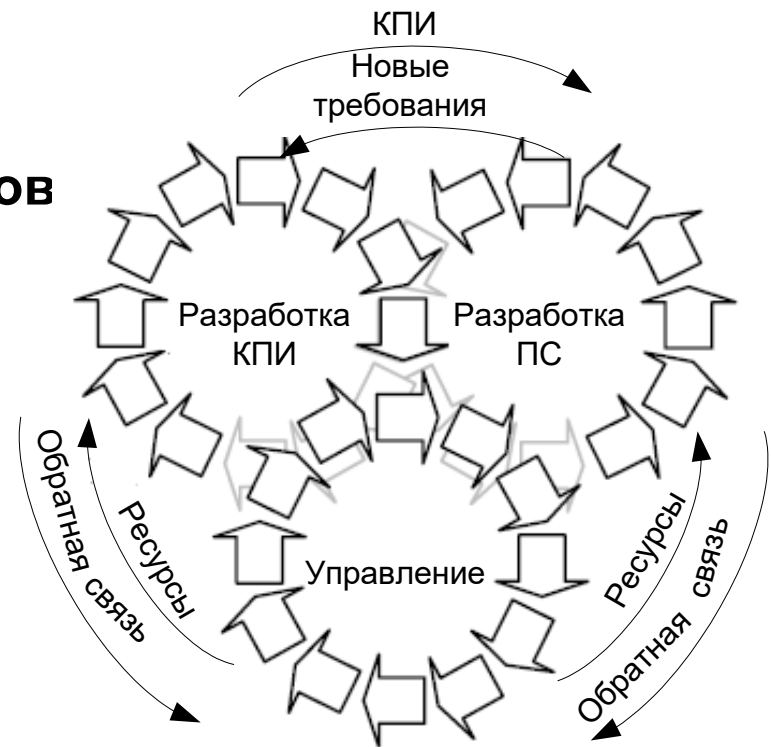
Метод сборки и библиотека интерфейса АПРОП (64 примитивов) реализованы в ОС ЕС (IBM 360), внедрены в комплексы РУЗА и ПРОМЕТЕЙ (В.В. Липаев, МНИИПА) для ВПК, переданы в 52 организации СССР и отмечен в Государственной премией СМ СССР (1985).

1. Глушков В.М., Лаврищева Е.М., Стогний А.А. и др. Система автоматизации производства программ (АПРОП), -Киев, ИК АН УССР, 1976.-134с
2. Лаврищева Е.М., Грищенко В.Н. Связь разноразличных модулей в ОС ЕС. – М.: Финансы и статистика.- 1982.- 136с.
3. Е.М. Лаврищева, В. Н. Грищенко "Сборочное программирование". К.: 1991.-213с.
4. В.В. Липаев, Б.А. Позин, А.А. Штрик "Технология сборочного программирования", М.:1992.-271 с.

Фабрика Product Lines:

Продуктовая линия (Product Lines) создана в институте программной инженерии (SEI) США.

Product line (линия продуктов) и Product family (семейство программных продуктов - СПП) определены в словаре ISO/IEC FDIS 24765:2009(E), Systems and Software Engineering Vocabulary как «группа продуктов или услуг, которые имеют общее управляемое множество свойств, которые удовлетворяют потребностям определенного сегмента рынка продуктов».



Формат линии

Системные AppFab в среде Интернет

В каждой ОС массового применения созданы фабрики программ. В библиотеках систем содержатся наборы документов, заготовок, КПИ, наборы данных и др. Они используются для сборки и интеграции из них приложений и систем для разных целей, включая управление предприятиями, большими данными и др.

К фабрикам системного типа относятся:

- 1) Фабрика ПО (Fabrics Software) для систем и сервисов;
- 2) AppFab в системе коллективной разработки VS.Net;
- 3) AppFab в системе Grid Европейского проекта;
- 4) AppFab IBM для создания доменов и бизнес-приложений;
- 5) AppFab CORBA для сборки разнородных модулей;

Фабрики ПО. Разработана группой r&r и включает коллекцию ресурсов, блоков кода, образцов приложений и т.п. [30]. В состав фабрики вошли принятые схемы и стандарты, рекомендации, инструменты, шаблоны решений и запуска новых приложений.

Архитекторы модифицируют требуемые коды сервисных ресурсов модели SOA Visual Studio Industry Partners (VSIP), а затем повторно устанавливает их на фабрике ПО. 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 находится в сообществе разработчиков фабрики ПО и GotDotN. В ее состав входят процессы предприятия, договора и пакет документаций и рекомендаций для приложения Global Bank агентами.

Благодарю за внимание